AD-A076 978   AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOO--ETC F/G 15/5
AN INVESTIGATION OF THE SOFTWARE REQUIREMENTS ALLOCATION PROCES--ETC(U)
SEP 79 V L COOPER
UNCLASSIFIED AFIT/GSM/SM/79S-2   NL

1 OF 3
AD
A076978

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

1

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER 14 AFIT/GSM/SM/79S-2	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) 6 AN INVESTIGATION OF THE SOFTWARE REQUIREMENTS ALLOCATION PROCESS IN THE ACQUISITION AND MANAGEMENT OF A MAJOR DEFENSE SYSTEM		5. TYPE OF REPORT & PERIOD COVERED MS Thesis 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) 10 Virgil L. Cooper, Capt., USAF		8. CONTRACT OR GRANT NUMBER(s) 12 282
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Institute of Technology(AFIT/EN) Wright-Patterson AFB, Ohio 45433		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS LEVEL II		12. REPORT DATE 11 21 September 1979 13. NUMBER OF PAGES 282
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) 9 Master's thesis,		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Approved for public release; IAW AFR 190-17

JOSEPH P. HIPPS, Major, USAF
Director of Information

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Acquisition Management	Requirements Analysis	Software
Configuration Management	Requirements Decomposition	System
Computer Programs	Requirements Allocation	Design
Selection Criteria	Horizontal Allocation	Weapon
Systems Analysis	Statistical Analysis	Systems

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The importance of computer hardware and software in the Department of Defense(DoD) has increased over the past 20 years to the point where computer technology is vital to the defense of our country. In addition, this technology has placed a tremendous strain on the fiscal assets of the DoD and Air Force (AF).

This thesis documents an investigation of the software requirements allocation process in the acquisition and (cont.)

012 225

JCB

over

20.

management of a major defense system. More specifically, this research identified the DoD and AF policies and procedures on CPCI selection, determined what criteria are currently used by system program offices(SPOs) and contractors to allocate system requirements to CPCIs, and evaluated the feasibility and potential impacts of an alternate approach to CPCI selection, called horizontal allocation.

An extensive literature review was performed, and a semi-structured interview questionnaire was executed to collect the research data. The questionnaire was designed to capture both objective and subjective data from a sample population that included 45 "software experts" from 10 AF organizations, 2 Federal Contract Research Centers, and 11 Aerospace contractors. Both qualitative and quantitative analysis techniques were used to interpret the interview results. The qualitative analysis consisted of categorizing, analyzing, and summarizing the answers expressed by the respondents. Three statistical techniques (Pearson Product-Moment Correlation, Chi-square test, and Student's t test) were applied to the quantitative data.

The results should lead to a better understanding of the CPCI selection process. DoD and AF policies on CPCI selection require that computer resources be managed as items of major importance during the system acquisition life cycle; and that computer hardware and software be specified and treated as configuration items. On most programs evaluated, functional modularity and previous experience were the primary considerations used to allocate system requirements into CPCIs. This resulted in significant problems with assessing development status, achieving system integration, completing meaningful tests, and documenting the system. On the other hand, the selection of CPCIs on the basis of horizontal allocation is feasible and promises to favorably impact the cost, schedule, performance, and management parameters normally associated with an acquisition program. This alternate approach is based on defining a software-intensive system in terms of system versions or models, each of which contains end-use system functional capabilities.

AN INVESTIGATION OF THE SOFTWARE REQUIREMENTS ALLOCATION PROCESS IN THE ACQUISITION AND MANAGEMENT OF A MAJOR DEFENSE SYSTEM

THESIS

AFIT/GSM/SM/79S-2 Virgil L. Cooper
Capt. USAF

AFIT/GSM/SM/79S-2

AN INVESTIGATION OF THE SOFTWARE REQUIREMENTS ALLOCATION PROCESS IN THE ACQUISITION AND MANAGEMENT OF A MAJOR DEFENSE SYSTEM

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University (ATC)
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science

by

Virgil L. Cooper
Capt. USAF

Graduate Systems Management
21 September 1979

Preface

To "fly and fight" effectively, the Air Force relies on modern weapon systems which are expensive and complex. These systems make extensive use of computer hardware and software to perform many functions which were previously performed manually, by hardware, or were not able to be performed at all prior to the advent of computer technology. As a result, management of the development and acquisition of software is a subject of increasing importance to the Air Force.

This subject is also important to me for two reasons. First, I experienced many of the frustrations and disappointments normally associated with software acquisition as a member of a major command, control, and communications(C^3) system program office in my last assignment. Secondly, my follow-on assignment is to a Computer Resources Acquisition Management Study Group at Headquarters Air Force. Hopefully, the results of this research will, in some small way, be helpful in solving the "software problem."

There are many people whose help and contributions made this research effort possible. I offer my sincerest thanks to all of them. I am especially grateful to all those "software experts" who assisted in collecting the research data. A special word of appreciation is extended to Mr. Charles E. Bobbish for his invaluable insight and advice during the past year.

I also express my deep appreciation and gratitude to my advisor, Professor Charles W. McNichols for his advice and encouragement throughout the research period. In addition, thanks are extended to my reader, Professor Alan A. Ross, for his invaluable assistance. This study would not have been possible without their help.

Most of all, I am deeply indebted to the Brooks family. Susan patiently strived for perfection while typing this thesis at all hours of the day; and Terry reformatted several tables and figures in order to present the results more clearly.

Finally, I express my undying warmth and affection to those special friends who provided moral support and encouragement during the darkest moments of the past 16 months.

Virgil L. "Lee" Cooper

Contents

List of Figures

List of Tables

Abstract

The importance of computer hardware and software in the Department of Defense(DoD) has increased over the past 20 years to the point where computer technology is vital to the defense of our country. In addition, this technology has placed a tremendous strain on the fiscal assets of the DoD and Air Force(AF).

This thesis documents an investigation of the software requirements allocation process in the acquisition and management of a major defense system. More specifically, this research identified the DoD and AF policies and procedures on CPCI selection, determined what criteria are currently used by system program offices(SPOs) and contractors to allocate system requirements to CPCIs, and evaluated the feasibility and potential impacts of an alternate approach to CPCI selection, called "horizontal allocation."

An extensive literature review was performed, and a semi-structured interview questionnaire was executed to collect the research data. The questionnaire was designed to capture both objective and subjective data from a sample population that included 45 "software experts" from 10 AF organizations, 2 Federal Contract Research Centers, and 11 Aerospace contractors. Both qualitative and quantitative analysis techniques were used to interpret the interview results. The qualitative analysis consisted of categorizing, analyzing, and summarizing the answers expressed by the respondents. Three statistical techniques (Pearson Product-Moment Correlation, Chi-square test, and Student's t test) were applied to the quantitative data.

The results should lead to a better understanding of the CPCI selection process. DoD and AF policies on CPCI selection require that computer resources be managed as items of

major importance during the system acquisition life cycle; and that computer hardware and software be specified and treated as configuration items. On most programs evaluated, functional modularity and previous experience were the primary considerations used to allocate system requirements into CPCIs. This resulted in significant problems with assessing development status, achieving system integration, completing meaningful tests, and documenting the system. On the other hand, the selection of CPCIs on the basis of horizontal allocation is feasible and promises to favorably impact the cost, schedule, performance, and management parameters normally associated with an acquisition program. This alternate approach is based on defining a software-intensive system in terms of system versions or models, each of which contains end-use system functional capabilities.

AN INVESTIGATION OF THE SOFTWARE REQUIREMENTS ALLOCATION PROCESS IN THE ACQUISITION AND MANAGEMENT OF A MAJOR DEFENSE SYSTEM

I. Introduction

Modern weapon systems[1] make extensive use of computers and software to perform functions which were previously performed manually, by hardware, or were not performed at all prior to the advent of computer technology. As a result, the importance of software in the Air Force(AF)[2] continues to intensify as new systems emerge in response to increasing threats and declining force levels. Thus, computer technology has become central to the Air Force's ability to perform its role and mission.

This technology has placed a tremendous strain on the fiscal assets of the Department of Defense(DoD) and the AF. In 1974, estimates of the annual Automatic Data Processing (ADP) costs in the DoD were $2.9 billion to $3.6 billion for software and a total of $6.2 to $8.3 billion when hardware

[1] In this thesis, the term weapon system refers to any DoD system that is not of a general purpose, commercially available nature. That is, it includes aircraft systems; space and missile systems; command, control, and communications(C^3) systems; and intelligence systems. A glossary of definitions for common terms used throughout this thesis is included as Appendix A.

[2] A listing of the acronyms and abbreviations used in this thesis is attached as Appendix B.

and ADP resources were included (Asch, Kelliher, Locher, and Connors, 1975 b:3-16). The software-hardware cost ratio was approximately 1:4 in 1955, while the 1985 ratio is projected to be 9:1 (Asch, et al., 1975 b:3-48). In fact, on one existing program, the World Wide Military Command and Control System(WWMCCS), the ratio is already in that vicinity (Myers, 1978:13). Given such ratios, it becomes increasingly important not only to cope with the technology but to master and exploit it, both from a technical point of view and, equally important, from a management and policy one.

The need to manage software as a critical component of a defense system over its life cycle is becoming widely recognized. A general awareness of this need as a defense-wide problem has been growing as software problems have reached top-level management visibility with increasing regularity. In July 1978, Barry C. DeRoze and Thomas H. Nyman of the Office of Secretary of Defense stated, "It is our opinion that DoD has been doing a poor job managing this increasingly important resource, and further, we have been doing little to encourage the application of science and technology to improve it. Both of these shortcomings must change!" (DeRoze and Nyman, 1978:309).

Background

In reflecting on a number of large, complex computer systems, Dr. Frederick Brooks of the University of North Carolina, described the development of large software projects as, "Like the mortal struggle of prehistoric beasts trying to escape the tar pits:

> Large systems programming has over the past decade been such a tar pit, and many great and powerful beasts have thrashed violently in it. Most have met goals, schedules, and budgets. Large and small, massive and wiry, team after team has become entangled in the tar. No one thing seems to cause the difficulty-any particular paw can be pulled away. But the accumulation of simultaneous and interacting factors bring slower and slower motion. Everyone seems to have been surprised by the stickiness of the problem, and it is hard to discern the nature of it. But we must try to understand it if we are to solve it" (Brooks, 1975:4).

For many years, the DoD has had its own version of the tar-pit problem. Difficulties have appeared in systems supporting all functional areas, including management support, command and control, intelligence, communications, and embedded computer systems. In fact, recent defense-sponsored studies have shown that most software development projects are unsuccessful in terms of performance, schedule, and cost. The final software product delivered to the user often deviates significantly from the original specification; delivery schedules are sometimes slipped for years; and cost overruns are often on the order of 200 to 300 percent (Ramamsorthy, 1975:46). Thus, one of the most critical periods of the software life cycle which needs additional management emphasis is the acquisition process.

The DoD policy on major system acquisition states, "Responsibility for the management of system acquisition programs shall be decentralized to the DoD components except for the decisions retained by the Secretary of Defense" (DoDD 5000.1, 1977:2). Thus, DoD components are responsible for a continuing analysis of their missions to identify mission needs and to define, develop, produce, and deploy systems to satisfy those needs. The system acquisition process, then, is a sequence of specified phases of program activity and decision points directed at the achievement of established program objectives.

The process is initiated with the approval of a mission need (DSARC 0) and extends through five major phases with an affirmative decision required as preliminary to proceeding into the second, third, and fourth phases (Ref. Figure 1). Thus, the system acquisition process is defined as

a. the conceptual phase, followed by a DoD program decision (DSARC 1),

b. the validation phase, followed by a DoD ratification decision (DSARC 2),

c. the full-scale development phase, followed by a DoD production decision (DSARC 3),

d. the production phase, and

e. the deployment phase (AFSCP 800-3, 1976:1-1).

The implementation of this process in the Air Force is normally delegated to the Air Force Systems Command(AFSC). Program Offices(POs) are then established in one of the product divisions (i.e., Electronic Systems Division, Aeronautical Systems Division, or Space and Missile Systems

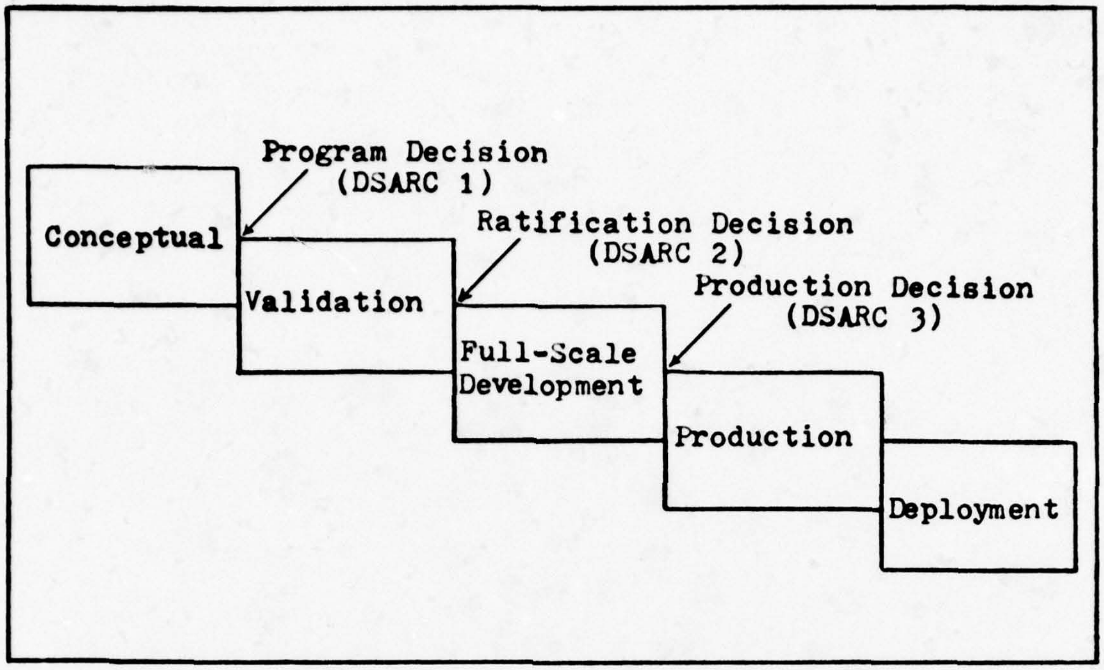


Figure 1. Systems Acquisition Life Cycle Phases

Organization) to perform the acquisition management function. This is accomplished by a complex set of interrelated, but separately identified, management disciplines. These disciplines are engineering, procurement, program control, test and evaluation, deployment, integrated logistics support, data management, and configuration management.

These disciplines represent separate areas of management responsibility which correspond, largely, with individual career specialities of PO personnel. They are also the basis for the classical PO organization and for major topics addressed in the various acquisition management regulations, specifications, and standards. However, two of these disciplines, engineering and configuration management, are intimately related in the system acquisition process. Engineering, as used here, refers to any or all of the various areas of technical effort involved in acquiring a DoD system. It is the general term which encompasses system engineering, equipment engineering, software engineering, and human factors engineering. Within this broad concept, the engineering discipline is responsible for requirements analysis, design studies, evaluation of performance, cost and schedule impacts of engineering change proposals(ECPs), deviations, and waivers, and technical direction to the developer. The products of these analyses, studies, and evaluations are the concern of configuration management.

Therefore, configuration management is a discipline that integrates the technical and administrative direction and surveillance to:

a. identify and document the functional and physical characteristics of a configuration item,

b. control changes to those characteristics, and

c. record and report change processing and implementation status (AFR 65-3, 1975:A2).

Thus, this discipline is categorized into three major areas of effort: identification, control, and status accounting. The first of these, configuration identification, is established in the form of technical documentation that becomes more detailed as development proceeds through the acquisition cycle. Three stages of configuration identification are generally employed during the development of a system. They are:

(1) Functional Configuration Identification(FCI). First, the FCI for the system is defined in the system specification or system-segment specification. Specifically, the specification prescribes all necessary functional characteristics, required performance parameters, the tests required to demonstrate compliance with the specification, the necessary system interfaces, and any design constraints.

(2) Allocated Configuration Identification(ACI). Later in the development process, the requirements of the FCI are allocated to individually-identified subsets called configuration items(CIs) for hardware and computer program configuration items(CPCIs) for software. These are normally identified during the validation phase as a part of the system engineering process. This description of "what" is to be done is documented in individual development specifications (called Part I Specification).

(3) Product Configuration Identification(PCI). Finally, the PCI describes in product specifications (called Part II Specification) the "as-built" configuration for CIs and CPCIs. This specification, along with technical orders and drawings, is the basis from which changes to the system are controlled after deployment.

(Airborne Systems Software Acquisition Engineering Guidebook for Configuration Management, 1978:12)

The subject of this research is the process of developing the allocated configuration identification for the system level requirements that are to be implemented via software. This process, called software requirements allocation, is of interest to me for two reasons. First, I experienced many of the frustrations and disappointments associated with software acquisition as a member of a major command, control, and communications(C^3) systems program office in my last assignment. During that four year period, I participated in all phases of the system acquisition cycle as a Computer Systems Analyst and a System Integration and Test Monitor. In those positions, I experienced several problems that probably could be attributed to the CPCI structure of the system. Secondly, my next assignment is to a Headquarters Air Force Acquisition Study Group. Hopefully, the lessons learned in my previous assignment and the results of this research will contribute to improvements in the DoD weapon systems acquisition procedure.

Previous Studies

The importance being placed on weapon system software acquisition and management by the DoD is reflected by the number of recent management and technical papers, guidebooks, conferences, and study groups either sponsored by the DoD or participated in directly by DoD personnel. As early as 1964, high-level Air Force managers had already recognized that computer technology was developing rapidly and that expanding requirements would dictate a proliferation of applications (Drezner, Shulman, Ware, Smith, Davis, Reinstedt, and Turn, 1976:3). Since then, several noteworthy studies have identified the need for a good system analysis of requirements and proper configuration management. For example, the 1972 study, CCIP-85, Information Processing/Data Automation Implications of Air Force Command and Control Requirements in the 1980's, identified requirements analysis/design/exercise technology as one of the five most critical problems needing further research and development. Specifically, the executive summary concluded that:

> "The nation's survival and prestige rest continuously on the assumption that incidents similar to the Pueblo and Liberty incidents would not occur during grave strategic confrontations. Information processing techniques could and should be doing more in the areas of C&C system requirements analysis, system design and system exercising to assure that this will not happen" (Asch, et al., 1975 b:3-4).

In another DoD sponsored study, there was a strong feeling of urgency that an energetic program of research be undertaken to advance the software art. Those in attendance at the 1973 Symposium on the High Cost of Software were in strong agreement that direct and indirect software costs are unnecessarily high and are growing rapidly. In part, they concluded that:

> a. "The key to achieving understandability appears to be structure, i.e., the partitioning of a program into parts and the organization of the parts into levels of abstraction ...,"
>
> b. application of any of the current methodologies (i.e., top-down design, modular decomposition, formal specification, etc.) would be unwise unless appropriately supported-both organizationally and technically" (Asch, et al., 1975 b:3-24).

In addition, an Ad Hoc Group of the Army Scientific Advisory Panel reported that many software problems are rooted in an inadequate initial system design effort. It recommended an orderly system design, considering all reasonable alternative system architectures before a development is initiated (Asch, et al., 1975 b:3-60). Dr. Barry Boehm drew a similar conclusion at the 1974 Aeronautical Systems Software Workshop when he presented a table that compared alternative software design techniques. He strongly emphasized the necessity of more work at the software design and structuring stage by presenting an analysis of errors on a large, good software project (Asch, et al., 1975 b:3-40). In another paper, entitled, "Specifications: A Key to Effective Software Development," the authors stated, "A technology needs to be developed that permits the structured decomposition of the system specification into a complete and consistent set of data processing subsystem requirements (Belford, Bond, Henderson, and Sellers, 1976:72).

Although these and many other papers have indicated a need for further research into system structuring and the process of software requirements decomposition, an indepth study of these topics was not discovered in a very extensive, but not exhaustive, literature search. However, requirements decomposition is addressed in a few papers that were presented at various conferences. For example, Kenneth G. Salter stated, "Many software errors are due to design rather than coding. The root of design errors is the lack of formal methodology for developing final code from top-level system requirements." He then proceeded to present a methodology for system decomposition representing data processing requirements in terms of functions, control, functional flows, and data. In conclusion, the author stated, "The methodology has been applied to two sample problems, suggesting that it may be adequate" (Salter, 1976:91-97).

In summary, the literature is replete with studies, technical papers, and conference proceedings that address problems associated with software engineering and software acquisition management. However, none appear to have addressed the process of allocating the system requirements into configuration items and computer program configuration items.

Statement of the Research Problem

Since weapon systems are not procured as a single identifiable system but rather as separate end items of contractor developed, federal supply stock, and commercial "off-the-shelf" items, a number of configuration items are normally identified for each system. Furthermore, the Air Force definition of a configuration item is, "An aggregate of hardware/computer programs or any of its discrete portions, which satisfies an end-use function, and is designated by the Government for configuration management" (AFR 65-3, 1974:24).

Thus, the number and composition of configuration items is a critical design issue since the Government's technical management activities primarily focus on CIs and CPCIs. For example, the developer is normally required to prepare and deliver individual performance specifications, test plans, test reports, product specifications version description documents, and users manuals to the Government for each CPCI. In addition, each CI and CPCI undergoes individual design reviews and configuration audits. Successful completion of these milestones are often misinterpreted by the System Program Office(SPO) as control of the development effort and visibility into the system development process. In reality, the Government has neither control nor visibility since these documents, reviews, and audits offer very little indication of the system's ultimate performance in an integrated environment.

This situation and the previously stated need for further research into the process of allocating system requirements into CIs and CPCIs define the problem area which is the subject of this research. Specifically, the thesis is an investigation of the present software requirements allocation process and an evaluation of an alternate method of selecting CPCIs.

Objectives of the Research

The overall objective of this study is to determine how the data processing requirements of a system are allocated to CPCIs and to investigate the feasibility of an alternate methodology. Specifically, the research will attempt to:

a. determine the DoD and AF policies and procedures for CPCI selection on a major system acquisition.

b. define what criteria are currently used by SPOs and contractors to allocate system requirements into CPCIs.

c. determine the advantages and disadvantages of those criteria.

d. determine how the selected CPCI structure impacts the overall acquisition management process.

e. determine the feasibility and evaluate the potential impacts of CPCIs defined in terms of system versions or models, each of which contains end-use system functional capabilities. This technique, called "horizontal allocation," would be developed on an incremental basis in which each "build" provides a system capability that is totally operational, rather than pieces and parts that are only partially operational.

Scope and Limitations of the Research

This study is limited to an investigation of the CPCI selection criteria used on large software-intensive weapon systems. In other words, information is not included for small systems that contain limited or no software, and for large systems where only one "natural" CPCI structure exists. Likewise, the large Automatic Data Processing(ADP) systems that are acquired under the Air Force 300-series regulations are excluded from this research. To include these would expand the scope of this research beyond the limited time and resources available. Therefore, the persons interviewed are primarily limited to Electronic Systems Division(ESD), Aeronautical Systems Division(ASD), Headquarters Air Force Systems Command(HQ AFSC), Federal Contract Research Centers (such as MITRE), and those contractors that are currently developing (or have in the recent past developed) large software-intensive systems.

This research will also be limited in that some sensitive information may not be available to the researcher, especially on existing programs that are experiencing problems. Similarly, the person(s) familiar with the criteria used for selecting CPCIs on a program may have departed the program and left no records.

Another limitation is the overall subjective nature of the interview instrument. In other words, most questions request the respondent's opinion and then ask why he has that opinion. Despite this constraint, some statistical analysis is possible, especially on the evaluation of the horizontal allocation methodology.

Finally, some of the interviews, especially with contractor personnel, were conducted by phone due to the limited time and resources available for travel to their different locations.

Assumptions

A significant amount of the data required for this research effort was collected by interviews, both personal and telephone. Therefore, an underlying assumption is that each person interviewed truthfully expressed his opinion of the situation on his program.

Thesis Organization

This thesis is organized into six chapters. This introductory section constitutes Chapter I. The methodology used to accomplish the research is described in Chapter II. The advantages of, disadvantages of, and rationale for selecting the methodology are described. Then, Chapter III presents a review of the present DoD and AF policies and procedures on the selection of CPCIs for a large software-intensive weapon system. In addition, the directives which govern the acquisition and management of computer resources during the development, acquisition, deployment, and support of major defense systems are summarized in order to establish the proper framework for discussing the CPCI selection process. The relevant studies, technical reports, proceedings of conferences, textbooks, and management guidebooks are reviewed in Chapter IV. Chapter V presents and discusses the interview results. Finally, Chapter VI contains a summary of and some conclusions on the process of software requirements allocation on a large software-intensive system. In addition, some recommendations on how the system acquisition process in the DoD could be improved are included.

II. Research Methodology

This thesis represents an investigation of one portion of the process of allocating system functional and performance requirements to individually identified subsets for purposes of managing their development. These subsets, called configuration items(CIs) for hardware and computer program configuration items(CPCIs) for software, are normally identified during the early stages of a system acquisition. The portion of this allocation process that was of concern in this research effort was the selection of CPCIs on a major weapon system that is software-intensive. More specifically, the research focuses on the criteria used in making the selection decision and the feasibility and potential impacts of an alternate methodology. Thus, the objective of this thesis was to answer the following questions:

a. What are the DoD and AF policies and procedures for CPCI selection on a major system acquisition?

b. What criteria are currently used by system program offices(SPOs) and contractors to allocate system requirements into CPCIs?

c. What are the advantages and disadvantages of those criteria?

d. How does the selected CPCI structure impact the overall acquisition management process?

e. Is horizontal allocation of requirements to be implemented via software feasible? Also, what are the potential impacts of defining CPCIs in such a manner?

In pursuit of the answers to these questions, a very extensive literature search and a semi-structured interview questionnaire were used for collecting the research data. Chapters III and IV contain the results of the literature search, and Chapter V presents and analyzes the interview results. The answers to these questions are then summarized in Chapter VI along with the attendant conclusions and recommendations. But first, the remainder of this chapter presents a brief description of the literature search and a detailed explanation of the methodology used to execute the interviews.

The Literature Search

The literature search was performed to develop a synthesized description of the software requirements allocation process currently in use. Policies, procedures, guidelines, criteria for allocation, and the associated impacts were typical categories of information searched for. Although the research topic was limited to the selection of CPCIs on a major weapon system that is software-intensive, over 300 references were identified and evaluated for applicability. Of those, 215 were selected as relevant to this research project. They included Congressional hearings; GAO reports; DoD and AF policies, regulations, pamphlets, and guidebooks; military standards; technical reports; conference proceedings; studies; textbooks; magazine articles; and "lessons learned." The majority of this published information addressed the acquisition and management of computer resources during the development, acquisition, deployment, and support of major defense systems. Thus, the problems, experiences, lessons learned, and alternative techniques associated with the CPCI selection process were often embedded in the discussions of requirements analysis, system design, software engineering, acquisition management, and software contracting.

The Interview Questionnaire

It is often claimed that the use of effective acquisition management techniques can significantly reduce the cost and development time for a major defense system. In the past 20 years, only a few systems have been delivered to the user within the original cost and schedule estimates. Therefore, two questions that often arise are: What is the root cause of these failures; and, what effect do the various management decisions have on the overall success of the program? To investigate one of those decisions (i.e., the CPCI selection decision), an interview questionnaire was designed to capture both objective and subjective data from Government and industry personnel experienced in the various disciplines of software development and management. In addition to appraising current and recent programs, these "experts" were asked to evaluate the feasibility and potential impacts of horizontal allocation of software requirements.

The initial interview questions were based on the researcher's four years of experience in the development and management of computer resources. During the development of the questionnaire, some questions were added and some were deleted as additional knowledge and insight were obtained through discussions with the research advisors. The result was a semi-structured questionnaire containing only 35 questions. However, by using "packing techniques" approximately 73 responses were possible. Although this number is not overwhelming, several of the questions solicited a specific response and then asked for supporting rationale in the form of a short narrative. The inclusion of these open-ended questions with multiple choice or fixed alternative types define the interview questionnaire as "semi-structured." In other words, the interview questions were standardized so that each respondent was addressing the same topics; and each respondent was provided an opportunity to present his subjective opinions on several questions. In addition, each individual interviewed was promised anonymity. To insure this anonymity, an individual and his organization are not identified with his comments.

The questionnaire was divided into three sections. Section I consisted of 11 individual demographic questions, such as organizational level of assignment, type of job, experience level, grade, educational level, number of years involved in system/software development, and number of system acquisition programs associated with. Section II contained two types of questions. First, the respondent was asked to identify the program he is presently (or was last) associated with and then to describe it in terms of its characteristics. For example, the questions covered total development cost, number of CPCIs, number of lines of code, cost ratio of software to hardware, and percentage of software development/contract that is complete. Then, several questions addressed the criteria used for allocating system software requirements to CPCIs on the identified program. Finally, Section III consisted of 5 questions that addressed the feasibility, perceived effectiveness, and implementation of horizontal allocation as an alternate technique for allocating software requirements to CPCIs. The perceived effectiveness question asked the respondent to evaluate horizontal allocation on the basis of its impact on 12 parameters. They were: debugging and testing efficiency, maintenance cost, morale of software experts, software integration, complexity of the decomposition process, training effectiveness, cost, development time, performance, management visibility, software quality, and early problem identification. Individual responses were scored on a simple 1 to 4 scale where 1 equals "not effective," 2 equals "somewhat effective," 3 equals "moderately effective," and 4 equals "very effective."

To eliminate ambiguity, a brief description of the horizontal allocation CPCI selection technique was included in Section III. The description included a definition and methodology for developing CPCIs selected by this alternate technique. In addition, an example of a simple Spacetrack System was attached to the questionnaire to explain the salient characteristics of horizontal allocation. This example, along with the questionnaire, are included as Appendix D to this thesis.

Method of Conducting the Interview

The interview questionnaire was designed, written, tested, and implemented in the spring and summer of 1979. The selection of potential respondents could have been accomplished through either a scientific, random method or a subjective approach. The random method was rejected for two reasons. First, the potential respondents are widely dispersed and with a time-consuming questionnaire, the number of people that could be interviewed had to be limited. Secondly, there was no means for identifying the population of "experts" who are presently (or were in the recent past) responsible for one of the various disciplines of software development and management. Therefore, a more subjective selection approach was chosen. Individuals were chosen who appeared to have special knowledge and experience involving software-intensive systems. Actual selections were based on the literature search, the researcher's experience, recommendations of the thesis advisor, and recommendations of individuals initially interviewed.

Initial contact was made in April and May of 1979 to determine if an individual was interested, willing, and able to participate. All of the various disciplines associated with software acquisition and management were included in the initial contacts. After many telephone calls, 87 of the 120 people contacted (72.5%) agreed to complete the questionnaire. They represented 10 Air Force organizations, 2 Federal Contract Research Centers(FCRC), and 11 Aerospace contractors. The number contacted, number that agreed to respond, and percent that agreed to respond are described in Table I for each of these categories.

TABLE I

RATE OF AGREEMENT TO RESPOND

	Number Contacted	Number Agreed to Respond	Percent Agreed to Respond
Government Personnel	72	42	58.3
FCRC Personnel	22	20	90.9
Contractor Personnel	26	25	96.2
Total	120	87	72.5

In addition to the questionnaire, each potential respondent was promised a cover letter that briefly outlined the purpose of this study, structure of the questionnaire, need for cooperation, and an estimate of the time required to complete the questionnaire. Since two types of interviews (personal and telephone) were initially planned, a standardized letter was written and mailed to all prospective respondents. The questionnaire was attached to the letters for those scheduled for telephone interviews (i.e., 57 of the 87 agreeing to respond). In other words, the researcher attempted to prevent general discussion or collusion among those to be personally interviewed by only providing the cover letter. This approach was abandoned after the initial interviews indicated that some forethought would have resulted in more accurate and comprehensive answers to some of the questions. Therefore, a second cover letter was written and forwarded, along with the questionnaire, to all those scheduled for personal interviews. Both of these letters are attached to this thesis as Appendix C.

Six of the 87 people agreeing to respond requested extra copies of the questionnaire for their subordinates, associates, or friends. As indicated in Table II, a total of 32 extras were distributed to government and contractor personnel. Therefore, a total of 119 questionnaires were distributed. Of those only 45 were returned by the cutoff date (1 August 1979). This resulted in an overall response rate

TABLE II

RESPONSE RATES FOR INTERVIEW QUESTIONNAIRE (Adjusted and Unadjusted)*

	Number Agreed to Respond	Total Number of Questionnaires Distributed*	Number of Questionnaires Returned	Percent Returned	Percent Returned (Adjusted)*
Government Personnel	42	56	18	42.8	32.1
FCRC Personnel	20	20	13	65.0	65.0
Contractor Personnel	25	43	14	56.0	32.6
Grand Total	87	119	45	51.7	37.8

* These numbers are adjusted to include the requests for additional copies of the questionnaire.

of 51.7% for those personally contacted by the researcher, and only 37.8% when the additional copies were included. In addition, 5 other people provided specific comments on the subject of the thesis, even though they did not complete the questionnaire. A listing of all respondents and their respective organization is included as Appendix E.

Although these rates appear to be low, especially in light of the close contact maintained between the researcher and those personally contacted, they are approximately the same as achieved on two similar studies. For example, the response rates for a Software Data Collection Study, performed by System Development Corporation(SDC) for the Rome Air Development Center(RADC) in 1976, were only 30% from internal SDC sources and 20% from military agencies (Willmorth, Finfer, and Templeton, 1976:1). Similarly, the University of California, Los Angeles Graduate School of Management distributed a Software Maintenance and Enhancement Questionnaire to 120 organizations and only received responses for 69 applications (Lientz, Swanson, and Tompkins, 1976:ii).

Accuracy of the Interview Questionnaire

The selection of a method for gathering data is one of the most important decisions a researcher will make, for the quality of his research depends not only on the adequacy of the research design but also on the quality of the data collection technique employed. Of the three basic methods available (i.e., controlled observation, document/record review, and survey), the survey provided the only practical approach for the objectives of this research. The four ways of conducting a survey are mail questionnaire, personal interview, panel interview, and telephone interview. As previously indicated, a combination of the mail questionnaire, telephone interview and personal interview techniques was used by this researcher. In other words, the sample population was identified, a questionnaire was mailed to the prospective respondent, and then a personal interview or telephone discussion was used to obtain answers to the survey questions. The selection of this approach was based on several criteria, such as factors to be measured and relevance, reliability, validity, and cost. They are each briefly discussed in the following paragraphs.

Factors to be Measured and Relevance. The type of information to be collected often influences the decision on which data collection method to use. Since the data to be captured included opinions and factual information, a survey type method was chosen as most appropriate for this research effort. In other words, the combination of telephone and personal interviews was judged to produce the most relevant data.

Reliability. To be useful, a data collection method and the rules for using the data must produce information that is not only relevant but correct. Two crucial aspects of correctness are reliability and validity. A method is reliable to the extent that it yields consistent results in repeated independent applications. This assumes that any random influence which tends to make measurements different on repeated applications is a source of measurement error (Nunnally, 1967:206). Time is not available for duplication of efforts which would identify the random influences. However, this questionnaire is believed to be reliable since no differences were detected between early and late responses.

Validity. It is possible for measurement scores to be highly reliable without being valid. That is, a questionnaire may be consistent without measuring what it is supposed to. But the converse is not true. Data cannot be highly valid without also being reliable (Nunnally, 1967:74). Validity, then, refers to whether an instrument measures what it is designed to measure. Determining the validity of some measurement instruments is rather easy. For example, verifying the proper performance of a yardstick as a measure of length is simple. However, validation of most instruments, especially those used to measure human feelings and behavior, is considerably more complex. Establishing validity for these instruments requires empirical investigations. Furthermore, the nature of evidence required for the validation depends on the type of validity being investigated.

Researchers are most often interested in four types of validity: concurrent, predictive, content, and construct.

An instrument that distinguishes individuals who differ in their present status is said to have concurrent validity (often referred to as discriminant validity). On the other hand, researchers use an instrument with predictive validity when they wish to distinguish individuals who will differ in the future. For some other instruments, validity depends primarily on the adequacy with which a specified domain of content is sampled. Content validity, then, requires a complete specification of the universe of possible responses and all the test items that might be used to measure it. Since the number of potential test items may approach infinity, determining content validity is frequently impossible. Finally, a researcher is often interested in determining a basis for inferring the degree to which an individual possesses some characteristic or trait. Since such characteristics usually cannot be identified with a single behavior, a variety of behaviors, called "constructs," are assumed to correlate with one another such that the characteristic can be identified. The process of validating this type of measuring instrument is called construct validity (Selltiz, Wrightsman, and Cook, 1976:171-179).

The type of validity this researcher primarily concerned himself with was content validity. Although empirical tests to demonstrate the validity of the interview questionnaire were not practical, it is assumed to provide valid data for several reasons. First, the questionnaire was reviewed, critiqued, and pretested by three software engineers assigned to the Directorate of Engineering at the Aeronautical Systems Division, Wright-Patterson Air Force Base, Ohio. Although these engineers are primarily concerned with the acquisition and management of computer resources embedded in aircraft systems, they are familiar with the techniques and issues associated with software development for various applications. As a result of their efforts, some questions were clarified and others were added.

Secondly, advance letters, refusal rates, sample selection, interviewer bias, and interviewee cooperation have all been demonstrated to have an affect on the validity of data obtained via personal and telephone interviews (Cooper, 1978:14-17). To reduce the adverse impacts, several precautions were taken during the data collection phase. For example, an individual was only included in the sample if he expressed an interest in the topic of the thesis. Then, explanatory cover letters (a type of advance letter) were mailed to each prospective respondent. These precautions in conjunction with the outstanding cooperation of those being interviewed and the researcher's efforts to remain unbiased support the validity assumption.

Cost. The cost associated with conducting face-to-face interviews often forbids administering personal interviews to a large sample population. In other words, the time and travel cost associated with questioning each individual separately is often prohibitive. Since telephone interviewing is generally considered to be significantly cheaper and to exhibit a high inter-interviewer reliability, a combination of the two interview methods was chosen for this research. This method appears to have resulted in data that is relevant, reliable, and valid.

Problems with the Interview Questionnaire and Methodology

Although much effort was expended trying to ensure the best designed, written, and implemented questionnaire, some problems were experienced. For example, only a small sample of the people and organizations developing and managing software-intensive systems was interviewed. Thus, one cannot conclusively state that the conclusions drawn in this thesis are true in all cases. Similarly, several questionnaires were returned with some questions unanswered. Several reasons can be hypothesized for this. They include: the level of detail was unknown to the respondent, the data was considered proprietary, the respondent's fear of reprisal (although anonymity was promised), and the respondent did not understand the question. This author concluded that the level of detail was unknown to the interviewee since "N/A" or "unknown" was written beside many questions. Furthermore, a sufficient number of responses was received to allow statistical analysis, even though the rate of blank responses was as high as 37% for a couple of highly technical questions.

The last problem experienced with the questionnaire concerned the definitions of "requirements allocation" and "requirements decomposition." Webster's New Collegiate Dictionary defines allocation as, "To apportion for a specific purpose or to particular persons or things: DISTRIBUTE tasks among human and automated components"; and decomposition as, "To separate into constituent parts or elements or into simpler compound" (Webster's New Collegiate Dictionary, 1976:30, 294). When applied to the requirements of a system specification and the act of defining components of the system, these terms have clear meaning. Unfortunately, this clear distinction is not projected in the literature, both regulatory and non-regulatory. As a result, a few of the respondents requested clarification before completing the questionnaire. In each case, allocation was described as the act of apportioning the system performance and functional requirements to individually-identified subsets for purposes of managing their development. The identification of configuration items was cited as the typical result of this apportioning process. On the other hand, decomposition was described as the hierarchial separation of system or subsystem requirements into units of development, such as functions, tasks, modules, subroutines, components and instructions. In each case, these explanations seemed to clarify the intent of the questionnaire. Therefore, the assumption of validity in the data appears to be a reasonable one.

Data Analysis Plan

Data analysis is an integral part of any research project. If survey data has been collected, the selection of an analysis technique is likewise very important. Since this thesis is highly management oriented, the primary objective in analyzing the data gathered in this research is to determine if an individual's demographics or the characteristics of the systems program he evaluated has any affect on his opinion of the utility of horizontal allocation. A variety of techniques are available for performing analysis on the data collected. Each technique chosen and the necessary preliminary steps are identified in Figure 2. Furthermore, they are all described in the following paragraphs, with one exception. In other words, the data collection step is not described in this section since it was discussed in the previous section entitled, Method of Conducting the Interview.

Type of Data. The type of data gathered in this research is, of necessity, both subjective and objective. Therefore, the questions must be separated into those requiring statistical analysis and those to be subjectively analyzed. Whenever possible, statistical methods were used since they provide assistance in understanding complex real-world processes and provide a standardized "language" for exchanging information about those processes among researchers and decision makers (McNichols, 1978:1-1). The respondent's answers to the questions that only lend themselves to

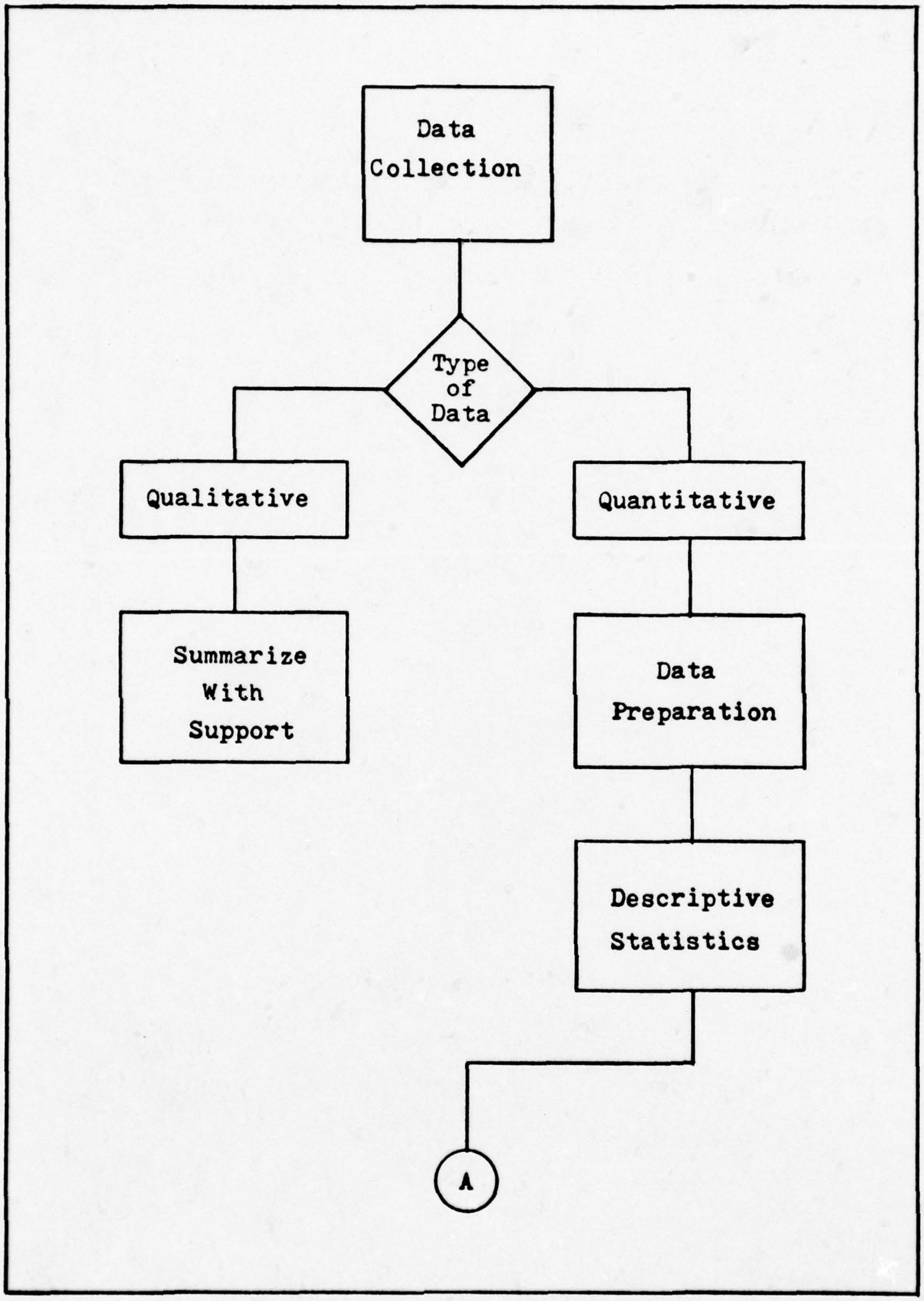


Figure 2. Data Analysis Plan

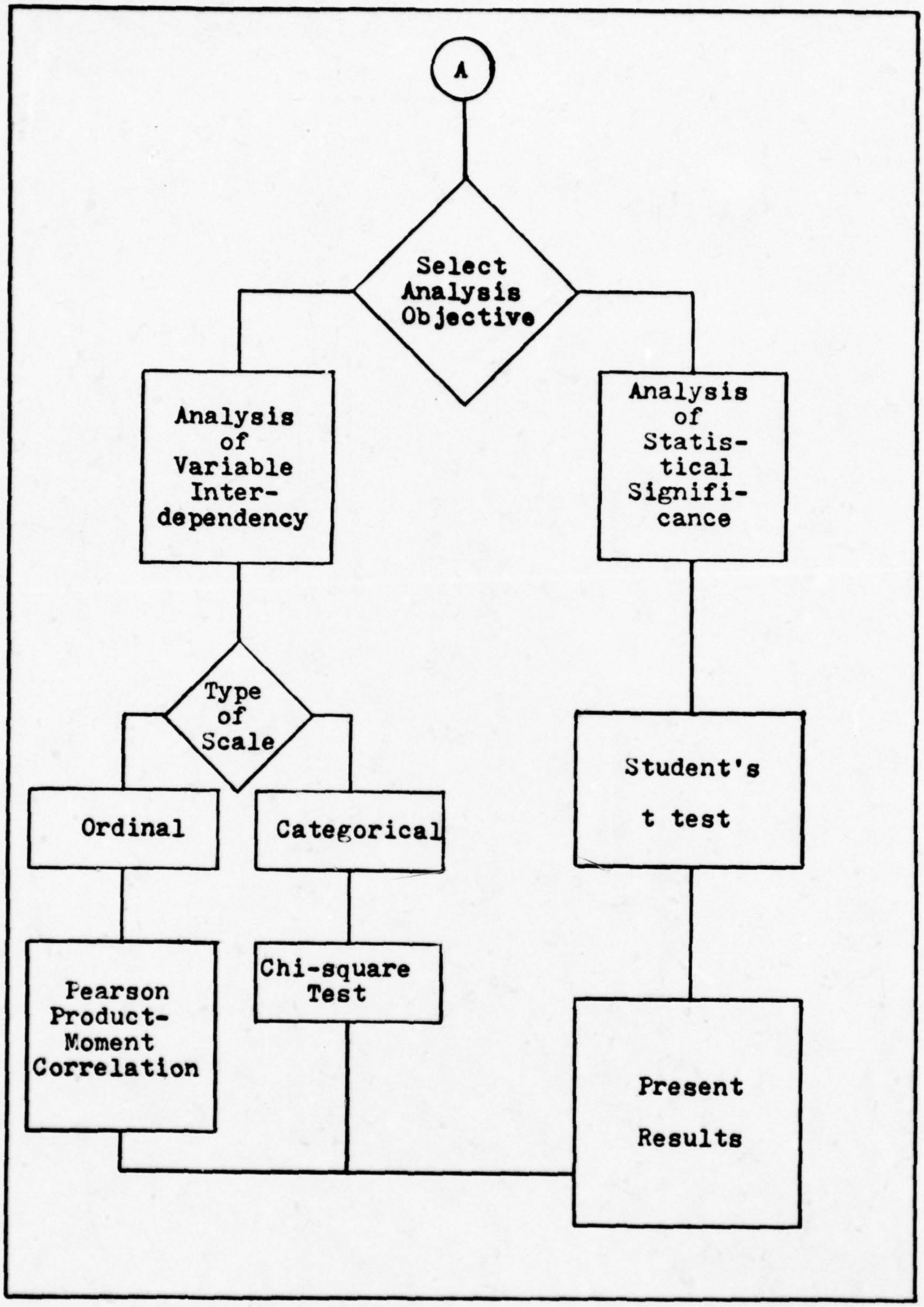

A
Select Analysis Objective
Analysis of Variable Inter-dependency
Analysis of Statis-tical Signifi-cance
Type of Scale
Ordinal
Categorical
Student's t test
Pearson Product-Moment Correlation
Chi-square Test
Present Results

Figure 2 (continued). Data Analysis Plan qualitative analysis are categorized, analyzed, and summarized in an attempt to reach some logical conclusions and, in some cases, recommendations. In addition, the summaries are supported with comments, experiences, and lessons learned. As previously indicated, the only identification attached to an individual's response is that it represents the opinion of a Government, FCRC, or Contractor expert.

Data Preparation. Upon receipt of each interview questionnaire, the answers to those questions requiring statistical analysis were coded and punched onto a standard IBM card. The data from a single respondent's questionnaire was punched onto a single card and is referred to as a case throughout the remainder of this thesis. Then, each case was edited for keypunch errors before being analyzed by the Statistical Package for the Social Sciences(SPSS) procedures. This integrated system of computer programs is designed to provide the researcher with a comprehensive set of automated statistical techniques commonly used in analyzing social science data (Nie, Hull, Jenkins, Steinbrenner, and Bent, 1978:1). The SPSS programs resident on the Aeronautical Systems Division CYBER 175 computer system at Wright-Patterson Air Force Base were used by this researcher for analyzing the interview data.

Descriptive Statistics. The basic distributional characteristics of each variable to be used in the subsequent statistical analysis were computed with the SPSS Frequency

Procedure. More specifically, information on the distribution, variability and central tendencies of each variable was obtained through selection of the "All" statistics and histogram options. In addition to the frequencies, this researcher is specifically concerned with the following univariate descriptive statistics: means, modes, and standard deviations.

Analysis of Variable Interdependency. After the descriptive statistics were studied, two analysis objectives were chosen as applicable to the data. The first one, analysis of variable interdependency, provides a measure of the strength of relationship between two variables. Two techniques were used for determining this relationship, depending on the type of scale used in measuring the data. They are the Pearson Product-Moment Correlation Coefficient for pairs of ordinal-scaled variables; and the Chi-square test for nominal-scaled variables (often called categorical data). These techniques were performed using the SPSS procedures PEARSON CORR and CROSSTABS, respectively.

Analysis of Statistical Significance. The second analysis objective was to determine the statistical significance of different responses to some of the interview questions. In other words, the objective was to determine if significantly more than half of the sample population agree that horizontal allocation is feasible and should be implemented on a future system. To make this determination, a one-tailed t test was used. There are two assumptions associated with that test. Both of them (random sample and normal distribution) are assumed to be satisfied.

Chapter Summary

This chapter presented a description of the research methodology used to investigate the process of allocating software requirements to CPCIs on a software-intensive system. In summary, an extensive literature search was used to ascertain the DoD and AF policies and procedures on CPCI selection on a major system acquisition. Then, a semi-structured interview questionnaire was designed, written, and implemented to investigate the CPCI selection process currently used by SPOs and contractors. In addition, the questionnaire was used to capture the perceived effectiveness of an alternate approach to CPCI selection, called horizontal allocation.

A description of the format of the questionnaire, method of conducting the interview, and accuracy of the questionnaire, is also included in this chapter. The questionnaire was divided into three sections: demographics, program characteristics and criteria currently used for allocating software requirements to CPCIs, and the feasibility and potential impacts of horizontal allocation. The questionnaire was reviewed and critiqued by three software engineers, and it was assumed to be basically reliable and valid. Since the questionnaire was designed with its accuracy in mind and no major problems were encountered in conducting the telephone and personal interviews, this assumption appears to be reasonable.

In addition, a comprehensive data analysis plan was outlined in Figure 2 and described in this chapter. In summary, two types of analysis (qualitative and quantitative) were used. The qualitative analysis consisted of categorizing, analyzing, and summarizing the answers expressed by the respondents. On the other hand, three statistical analysis techniques were applied to the quantitative data. They were Pearson Product-Moment Correlation, Chi-square test, and Students t test. Additionally, univariate descriptive statistics for each of the variables were obtained via the SPSS Frequency Procedure and the appropriate options.

III. DoD and Air Force Policies and Procedures

The acquisition and management of computer resources within the Federal Government are controlled by a complex hierarchy of policies and directives. At the top of that hierarchy is Public Law 98-306 which provides the basic structure and concepts for the government-wide system of Automated Data Processing Equipment(ADPE) management. In the Executive branch, this law has been implemented through the issuance of special rules by the Office of Management and Budget(OMB), the General Services Administration(GSA), and the individual federal agencies. To implement these rules, the DoD has issued a series of directives, instructions, and manuals, and the AF has published a series of regulations, pamphlets, and guidelines dealing with computer resources. In addition to these specialized rules, the acquisition and management of computer resources are also subject to the general rules covering all federal procurement and property management.

This chapter presents an overview of the regulatory structure which governs the acquisition and management of computer resources during the development, acquisition, deployment, and support of major defense systems. More specifically, the DoD and AF policies and procedures on the selection of CPCIs for a large software-intensive weapon system is the primary concern of this chapter. For example, a command, control, and communications(C^3) system is a large software-intensive weapon system, and it can inherently be partitioned into more than one set of CPCIs. Before discussing the DoD and AF policies and directives governing this process, a brief summary of how they evolved and how they interface with other governmental directives is required for proper orientation. Figure 3 provides an overview of these directives and how they interact (Kuo, Durieux, Forsythe, Leary, Marciniak, Putman, and Spiro, 1977:Atch 1).

Historical Perspective: ADPE Versus ECS

In the 1960's, the primary concern for computer resource management was the proliferation of computers. At that time, computers primarily automated functions that had been accomplished manually (e.g., payroll, supply). They were acquired to accomplish specific tasks and were viewed as special purpose machines. But there was a great similarity among types of user functions, resulting in the application of computers across functional lines. This was the beginning of the "General Purpose" computer. The resulting growth of automated applications was undisciplined and sometimes chaotic. During this period, under the auspices of legislation called the "Brooks Bill" (later passed as P.L. 89-306), and subsequent DoD Directives, authority was delegated to the Assistant Secretary of Defense(Comptroller) and the Secretary of the Air Force for Financial Management

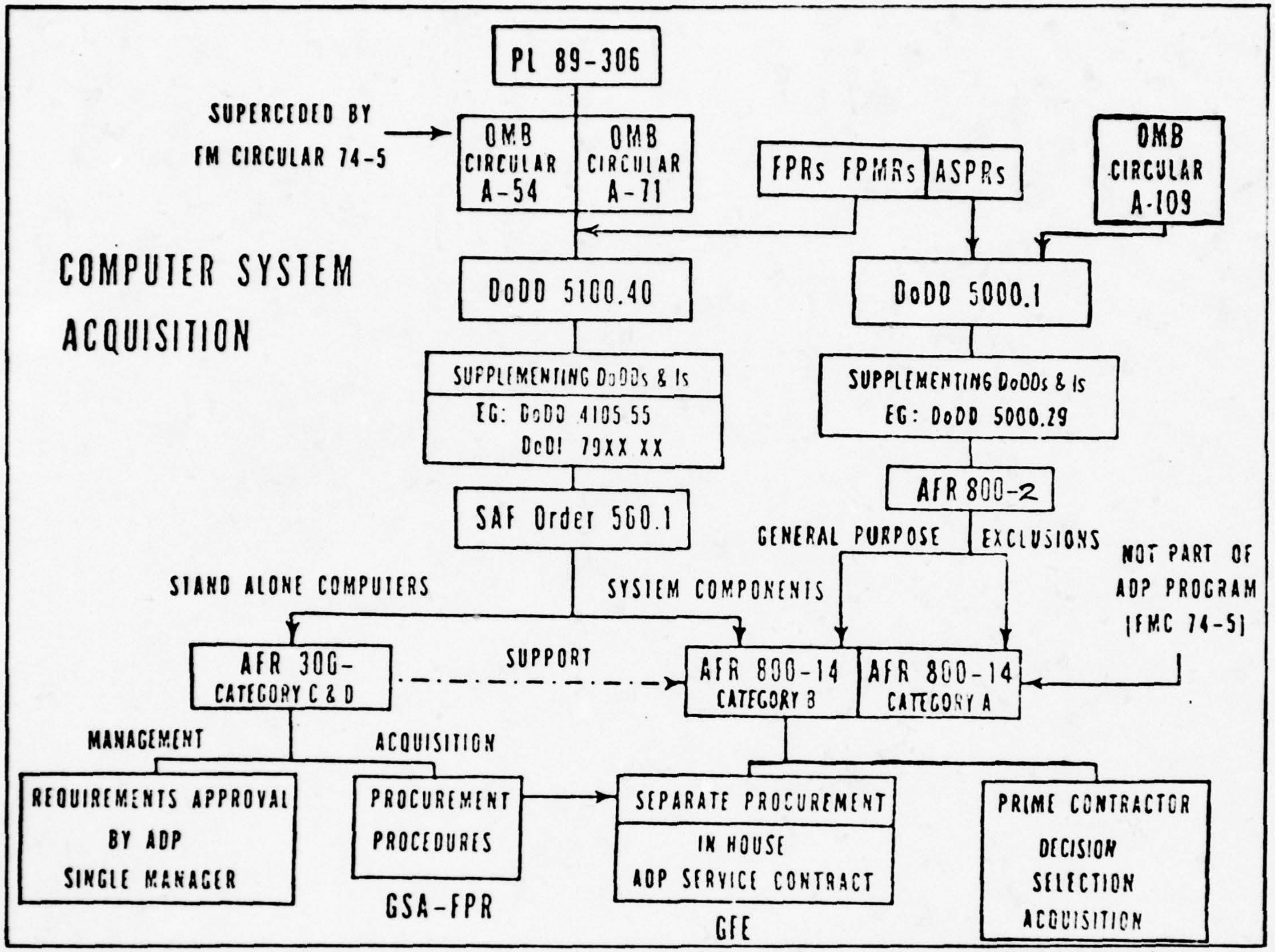


Figure 3. Hierarchy of Policies and Directives (Kuo, 1977:Atch 1)

(SAF/FM) to control the purchase of the off-the-shelf commercially available computers. The objective of this approach was to prevent costly duplication and waste of automatic data processing(ADP) resources (Kuo, et al., 1977:C-1).

At the same time, technology progressed to the point where small computers were installed as embedded components in most major defense systems. This special subset of computer applications are called Embedded Computer Systems(ECS) to distinguish them from ADP systems. They are physically incorporated into a larger system whose primary function is not data processing and their outputs generally include control signals, as well as computer data. In other words, they are integral to a weapon system from a design, procurement, and operations viewpoint. So, embedded computers range from small units in airborne systems to large units in ground and ship based command and control systems (DeRoze, 1977:1-1). Therefore, excluded from embedded systems are general purpose, commercially available ADP resources as defined and administered under OMB Circular A-71, DoD Directives 5100.40, 4105.55 and 4160.19, and the AF 300-series regulations (Ref. Figure 3).

This led to the evolution of a separate set of policies, directives, and regulations, such as OMB Circular A-109, DoD Directives 5000.1 and 5000.29, and AF Regulation 800-14 (Ref. Figure 3). The purpose of this series is to establish guidelines for the acquisition and support of computer equipment and computer programs employed as dedicated elements, subsystems, or components of major systems developed or acquired under the program management concept. Since the governmental ADP and ECS regulatory system is extraordinarily complex, it is beyond the scope of this chapter to present a complete and definitive review of all the policies, regulations, and procedures. Rather, a summarized and simplified overview of the more pertinent ones is provided. Particular emphasis is placed on those regulations that are concerned with the allocation of system requirements to CPCIs and how computer resources are managed during the acquisition of a major defense system. In other words, general-purpose commercially available ADPE resources are excluded. In addition, significant conflicts in the literature are highlighted.

Congress: Public Laws Affecting ADPE and ECS

Public Law 89-306. Numerous laws enacted by Congress have an impact on the acquisition and management of ADPE and ECS. The most important of these is P.L. 89-306. In 1965, Congress passed P.L. 89-306 to correct the pervasive mismanagement of ADPE in the federal government. Between 1958 and 1965, the Comptroller General submitted more than 100 audit reports to Congress showing a pattern of ADPE mismanagement. Government agencies acquired computers independently without regard to government-wide needs. As a result, the government did not receive any volume price discounts on its purchases although it was the largest computer user in the world. There was little computer sharing between agencies and many computers were under utilized. Despite these and other widespread problems, little action was taken by the Executive branch beyond the issuance of advisory guidelines. In Congress, the House Government Activities Subcommittee, under Chairman Jack Brooks, held hearings on Federal ADPE management in 1963 and 1965. The result was the passage of Public Law 89-306, commonly known as the Brooks Act, in October 1965 (Self, 1978:55-56).

The stated purpose of P.L. 89-306 is "... to provide for the economic and efficient purchase, lease, maintenance, operation, and utilization of automatic data processing equipment by federal departments and agencies. The intent of Congress was to achieve a businesslike Government-wide coordinated management effort ..." by providing a "delineation of responsibilities and stronger organizational plan for Government ADP management ..." (Senate Report No. 938, 1965:3877). Thus, P.L. 89-306 did not establish specific procurement or administrative policy. Rather, it established a centralized ADP management structure in the Executive branch and assigned responsibilities for providing ADP management.

Three agencies, OMB, GSA, and the National Bureau of Standards(NBS), were assigned specific responsibilities. OMB was assigned responsibility for policy and fiscal matters. GSA was assigned operational responsibility for ADPE procurement, maintenance, and utilization decisions. NBS was assigned the technical aspects of ADP management. Finally, certain responsibilities were left with the agencies which utilize ADPE, hereafter referred to as the user agencies (P.L. 89-306; Senate Report No. 938, 1965:3861, 3877-3885).

An exception to this centralized procurement is allowed by paragraph (b)(2) of the law. It permits GSA to delegate the lease, purchase, or maintenance of individual ADP systems to the user agency when necessary for reasons of economy, efficiency, national security, or national defense. In the implementation of P.L. 89-306, those computer resources integral to weapon systems and specially designed computer equipment were excluded from control. This specific exclusion was contained in OMB Circular 74-5 and DoDD 5100.40 (Ref. Figure 3). The purpose of this exclusion was to retain the authority of the Systems Program Manager to accomplish his assigned tasks. The impact of this exclusion and the resulting regulations are summarized in later paragraphs.

Armed Services Procurement Act of 1947. Other laws are more general and apply to all types of property, including computer resources. For example, the Armed Services Procurement Act of 1947 authorized the DoD to develop the Armed Services Procurement Regulations(ASPR). While the directives implementing P.L. 89-306 are essentially concerned with internal government policies, the ASPR deals primarily with the procurement and contracting relationship between the government and private enterprise. However, one source of considerable confusion and conflict in the acquisition and management of computer resources is the ASPR treatment of computer software. In summary, software is considered as an item of data in the same manner as reports, forms, manuals, and specifications are. More specifically, data management practices in DoD do not restrict data to items written or printed on paper; they include information recorded in suitable form on any suitable medium, such as film, photographic paper, magnetic paper or tape, and in digital or analog form (Searle, 1977:111).

This concept was reinforced in November 1974 when a revision of the ASPR appeared providing extensive new and revised coverage in Section IX on rights associated with computer software. The Defense Procurement Circular(DPC) 74-3 states that computer software falls within the definition of data, but is specifically excluded from the definition of technical data. As an item of data, ASPR requires the listing of computer software on the Contract Data Requirements List, DD Form 1423, as a measure to protect the Government's right in data. Although the ASPR appears to have a basis in legal decisions which have been reached in determining whether computer programs are things to be copyrighted and/or patented, the Air Force Systems Command(AFSC) has initiated an attempt to have the ASPR committee reverse its policies (Searle, 1977:111-112). This decision was based on AFSC's policy that computer software acquisition be subjected to the same contractual and cost controls as hardware. In other words, software delivery, like hardware, should be listed in the contract schedule with desired performance characteristics spelled out in the Statement Of Work(SOW). Furthermore, software is delivered as an active system component and functionally performs as part of a system. In this respect, software is the same as hardware; whereas, the documentation for hardware and software is rightfully data since it is supportive to the functioning of the system (AFSCRP 800-1, 1975:3).

Office of Management and Budget

The Office of Management and Budget(OMB) has broad policy and fiscal authority within the Executive branch. This authority derives from two sources. First, as a staff office of the President, OMB inherently possesses extensive authority to implement Presidential policy. Secondly, OMB has been assigned specific duties and responsibilities under certain laws, like P.L. 89-306. This policy and fiscal authority is traditionally implemented through the issuance of OMB circulars. Several circulars which directly address the procurement and management of ADPE resources have been issued.

However, only one, OMB Circular A-109, directly impacts the acquisition and management of embedded computer resources. This circular establishes policies to be followed by the Executive branch agencies in the acquisition of major systems. More specifically the circular covers all the acquisition management activities from analysis of agency missions to the successful introduction of the system into operational use. In addition, major system acquisition programs are defined as those that:

a. are directed at and critical to fulfilling an agency mission,

b. entail the allocation of relatively large resources, and

c. warrant special management attention.

The definition of additional criteria (including establishing relative dollar thresholds) for the determination of major systems is at the discretion of the Executive agency head (Glore, Friedman, and Goheen, 1978:42).

DoD Directives

Annual expenditures by the DoD on the design, development, acquisition, management, and operational support of computer resources embedded within and integral to weapons, communications, command and control, and intelligence systems are measured in billions of dollars (Asch, et al., 1975 b:3-16). At the same time, such computer resources have often presented critical cost and schedule problems during the development and acquisition of new defense systems. Even after system deployment, the software has often proven unreliable. To correct these problems and to improve the management of embedded computer resources in general, new policies and procedures have been implemented via DoD Directives. A brief summary of the directives applicable to the acquisition and management of computer resources in a major defense system follows. These summaries are provided to establish the framework for describing the Air Force policies and procedures on the allocation of system requirements to CPCIs in the acquisition of a large software-intensive system.

DoDD 5000.1. The Secretary of Defense(SECDEF) issued DoDD 5000.1, "Major System Acquisitions," to implement the policies of OMB Circular A-109. Specifically, this directive establishes policy for the management of programs designated as major system acquisitions. In addition to the criteria of "criticality" and "special management attention," relative dollar thresholds are specified for the designation of a major system acquisition. That is, system programs involving anticipated cost of $75 million in research, development, test and evaluation(RDT&E) or $300 million in production are designated as major system acquisitions. The directive also provides a guide for management of less-than-major system acquisition programs. Basically, the policy outlines the system acquisition process including responsibilities of the SECDEF, DoD components, the OSD staff, and the Defense Acquisition Executive. Key decision points and other major aspects of acquisition management are defined (DoDD 5000.1, 1977:1-3).

DoDD 5000.2. DoD Directive 5000.2, "Major System Acquisition Process," defines the policies and procedures for DoD activities in support of the Secretary of Defense's decision-making process on major system acquisitions. This directive supplements DoD 5000.1 and establishes the Defense Systems Acquisition Review Council(DSARC) charter. It further defines the responsibilities of DoD component heads in the identification and analysis of mission needs, and outlines scheduled program reviews. In addition, major program documentation, such as the Mission Element Need Statement(MENS) and the Decision Coordinating Paper(DCP) are described. Therefore, DoDD 5000.2 is a key directive for the acquisition of embedded computer systems (DoDD 5000.2, 1977:1-5).

DoDD 5000.29. The ever expanding applications and associated problems of embedded computer systems have become a concern at the highest levels of the DoD. As a result, new policies for the management of embedded computer resources have been developed. The Air Force has led the effort to establish controls and cognizance over the ECS area by publishing AFR 800-14, "Management of Computer Resources in Systems," which was followed by DoDD 5000.29, "Management of Computer Resources in Major Defense Systems." These documents recognized the need to have different but complementary instructions for the overlapping areas of embedded systems and general-purpose, commercially available automated data processing systems (Kuo, et al., 1977:C-2).

DoDD 5000.29 was therefore written to establish policy for the management and control of computer resources that are part of a major system acquisition as defined in DoDD 5000.1. These principles may be applied to other systems, with the exception of general-purpose, commercially available automated data processing systems which are administered under DoD Directives 4105.55 and 5100.40. Therefore, computer resources are defined for the purposes of this paper, as components which must be managed as elements or subsystems of major importance during conceptual, validation, full-scale development, production, deployment, and support phases of the life cycle. Particular emphasis on computer software and its integration with the surrounding hardware is also required (DoDD 5000.29, 1976:2).

This directive also establishes a Management Steering Committee for embedded computer resources at the Office of Secretary of Defense(OSD) level to oversee the implementation of the directive and the incorporation of its policies into the defense system acquisition process. In addition, policy is defined for the following major areas:

a. Requirements validation and risk analysis.

b. Configuration management of computer resources.

c. Computer resources life cycle planning.

d. Support software deliverables.

e. Milestone definition and attainment criteria.

f. Software language standardization and control.

(DoDD 5000.29, 1976:2-4)

In addition to the emphasis placed on computer resources, the configuration management policy established in this directive is of primary importance to the research effort. Therefore, the latter policy is quoted as follows:

> Configuration Management of Computer Resources. Defense system computer resources, including both computer hardware and computer software will be specified and treated as configuration items. Baseline implementation guidance for this action is contained in DoD Instruction 5010.21 (DoDD 5000.29, 1976:2).

This directive is not without conflict, however. For example, the use and definition of related terms, such as computer software, computer data, computer program, and software engineering have been interpreted:

a. as indeed sufficiently loose that their real intent is ambiguous in a number of respects, and

b. such that they may well prove to be in significant conflict with established Air Force practice (Searle, 1977:115).

DoDD 5000.19. This directive, entitled "Configuration Management," establishes policy for configuration management for all Military Departments, DoD components at all echelons, and all Defense-Industry interfaces. In summary, configuration management must be applied to all configuration items procured, or obtained by agreement between in-house activities. The directive describes responsibilities for initiation, planning, documentation and audit of configuration management. In addition, the processes of functional and allocated configuration identification, control, and status accounting are described (Glore, et al., 1978:44-45).

DoDI 5010.21. DoD Instruction 5010.21, "Configuration Management Implementation Guidance," provides guidance for DoD components in the implementation of Department of Defense policies on configuration management. It specifies that configuration management applies to all systems, equipment, and other designated material items. More specifically, the process should be tailored to the particular configuration item whether it is developed at government expense or privately developed and offered for government use. In addition, the instruction addresses and defines guidelines related to the following concepts of configuration management:

a. configuration identification,

b. configuration control,

c. configuration status accounting,

d. configuration audits,

e. procurement aspects,

f. logistic support aspects, and

g. implementation (Glore, et al., 1978:46).

Policies, regulations, and procedures implementing the first of these concepts are a major concern of this thesis.

Therefore, the directives that address the process of allocating system level requirements to CPCIs are summarized in the following sections.

Air Force Regulations

Recent emphasis in computer software acquisition has prompted the publication of numerous policies, regulations, standards, and guidebooks in the Air Force. Some of these have increased the emphasis that managers and users of systems must place on the computer resources. Others have clarified and broadened organizational responsibilities and authority. For example, using and supporting commands are required to accept an early role in planning for the operational and support requirements of software. However, the success of new policies rests not on their publication, but on a full understanding of the concepts underlying them and their proper implementation. Therefore, the most significant directives affecting the acquisition and management of software are summarized below. Particular emphasis is placed on those establishing policies and procedures for CPCI selection.

AFR 800-2. This regulation, entitled "Acquisition Program Management," implements DoDD 5000.1 and DoDD 5000.2 by establishing policy, responsibilities, and reporting requirements for major systems and for other Air Force acquisition programs. Maximum authority and responsibility for each program are delegated to the implementing command.

In addition, a single person known as the Program Manager (PM) is directed to plan, organize, and conduct the acquisition within the Air Force approved limits of system performance, schedule, and funding. Therefore, the Program Manager is the technical and administrative focal point for all program activities, including the participation of all other organizations (AFR 800-2, 1977:1-2).

AFR 800-14. The most important regulation affecting ECS software acquisition is AFR 800-14, Volumes I and II. Along with Air Force Systems Command supplement 1 to Volume I, this regulation establishes policy and provides guidance for planning, developing, acquiring, supporting, and using computer resources in systems acquired and managed under the AFR 800-2 program management concept. Thus, computer resources are the only commonly used components of Air Force weapon systems whose development is addressed in a separate regulation. This is partially because computer technology is relatively new and not well understood. In addition, software is on the critical path of many procurement efforts, and is therefore the root cause of many costly delays in system acquisitions.

Volume I, entitled "Management of Computer Resources in Systems," establishes policy for computer resources employed as dedicated elements, subsystems, or components of systems acquired and managed under AFR 800-2. More specifically, the Air Force policy on management of computer resources in systems states that:

a. Computer resources in systems are managed as elements or subsystems of major importance during conceptual, validation, full-scale development, production, employment, operation, and support phases. System performance requirements are allocated to subsystems using in-depth trade-off studies and cost-effectiveness analyses.

b. Configuration management procedures are developed to assure control during development, test, transfer and turnover, operational maintenance, and major modification.

c. Program Management Directives(PMDs) require and Program Management Plans(PMPs) provide for the specification and allocation of system performance and interface requirements to be met by computer equipment and computer programs.

d. PMDs require and PMPs provide for the identification of computer equipment and computer programs as configuration items.

(AFR 800-14, Vol. I, 1975:1-2)

In addition, Volume I assigns implementation responsibilities to Headquarters U.S. Air Force, Air Force Systems Command(AFSC), Program Manager, Air Force Logistics Command (AFLC), Using agencies, Air Training Command(ATC) and the Air University (AFR 800-14, Vol. I, 1975:2-3).

Volume II of AFR 800-14, entitled "Acquisition and Support Procedures for Computer Resources in Systems," consolidates procedures that apply when implementing the policies of AFR 800-14, Volume I and other related publications. It provides a centralized source document for the policies, procedures, and guidance required to manage the acquisition and support of computer resources in systems. The basis for this volume is "the uniqueness of computer resource management requires the publication of a document which consolidates and explains the applicability of other publications to computer resource acquisition and support" (AFR 800-14, Vol. II, 1975:1-1).

The specific areas this regulation addresses are as follows:

a. Computer resources in the system acquisition life cycle.

b. Planning.

c. Engineering management.

d. Computer program quality assurance.

e. Configuration management.

f. Documentation.

g. Contractual aspects.

h. Transition and turnover.

i. Support.

j. Computer program validation and verification(V&V).

(AFR 800-14, Vol. II, 1975)

Since the "other related publications" (referred to above) that are applicable to this thesis are summarized in later paragraphs, a detailed description of each area is not included here. However, the computer program life cycle defined in this regulation is described since it varies somewhat with that normally found in textbooks and other non-government publications. In addition, the interrelation of the system acquisition life cycle and the computer program life cycle are described below.

Computer program development can be conceptualized as the computer program life cycle shown in Figure 4 (AFR 800-14, Vol. II, 1975:2-5). The primary difference between this and the non-government definition is the Air Force treatment of software installation as a separate phase. The rationale for this appears to be the need to place special emphasis on the peculiar adaptation of the software to various sites for multi-site systems. In other words, the computer programs should be demonstrated to operate with a specified level of confidence at each site although they have previously been successfully qualified and integrated.

The interrelationships of the system acquisition life cycle and the computer program life cycle can vary depending on the specific program under development. The computer program life cycle may span more than one system acquisition life cycle phase or occur in any one phase. For example, a mission simulation computer program may undergo all of the phases of the computer program life cycle during the conceptual phase; while a mission application program may undergo the computer program life cycle phases during the validation, full-scale development, and production phases. In either case, the computer program life cycle, and the formal activities associated with it (configuration management, technical reviews, testing, audits, etc.), will occur at least once for each CPCI during the system acquisition life cycle. The activities need not be sequential; instead, there are

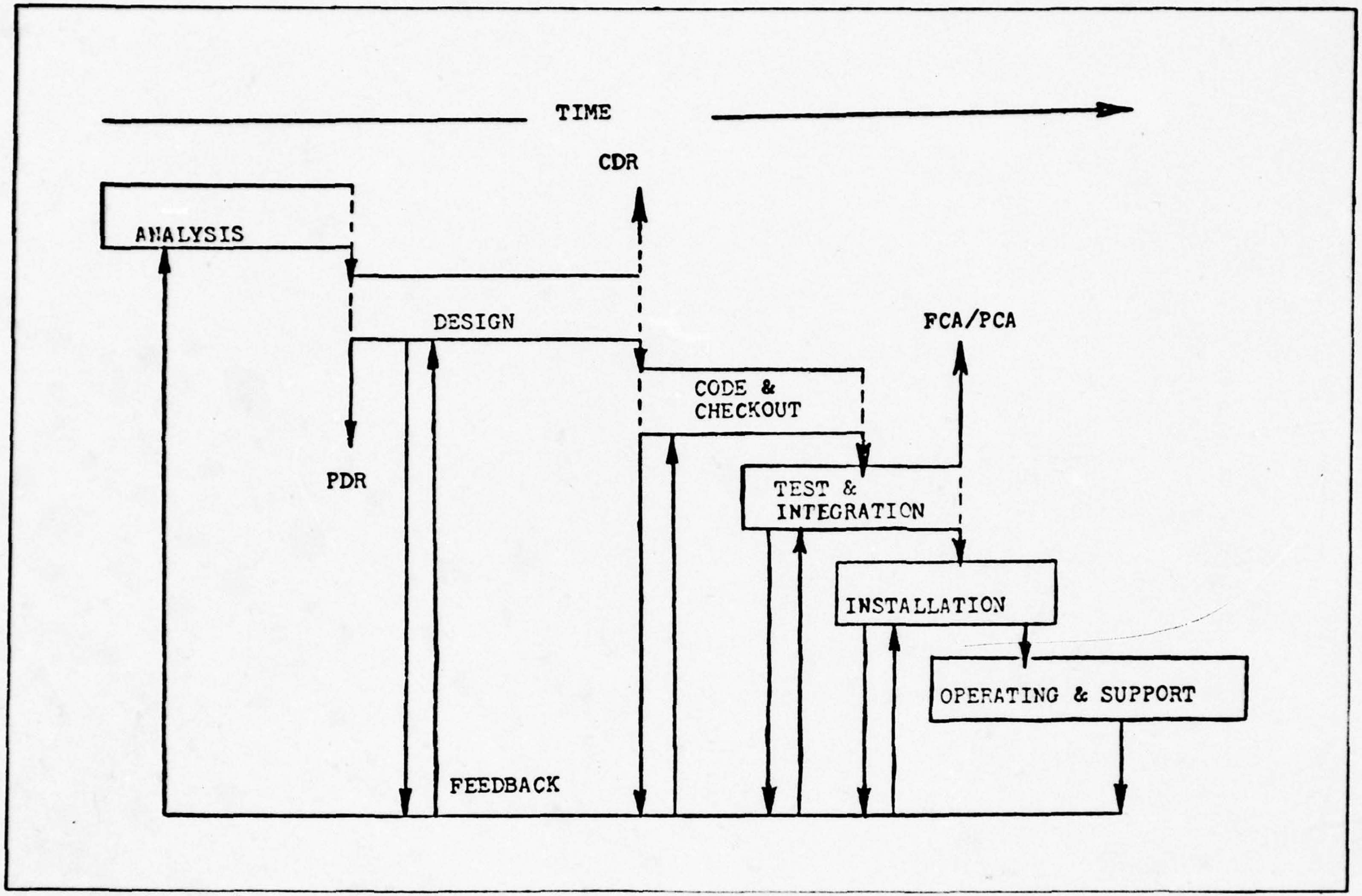


Figure 4. Computer Program Life Cycle (AFR 800-14, Vol. II, 1975:2-5)

potential loops between all phases. For example, design may reveal problems which lead to the revision of requirements and reinstitution of certain analyses (AFR 800-14, Vol. II, 1975:2-3).

The DoD Directives and Air Force Regulations covered to this point have been primarily concerned with the acquisition and management of computer resources in systems. They were summarized to establish the overall policies on computer programs. Within that general framework, the specific policies, concepts, techniques, and procedures for CPCI selection on major system acquisitions (as prescribed by AFR 65-3, and AFSC Pamphlets 800-3 and 800-7) are described below.

AFR 65-3. This joint DoD Services/Agency regulation, entitled "Configuration Management," prescribes uniform policies and guidance for the Military Services and Defense Agencies (hereafter referred to as DoD components) responsible for implementation of configuration management within the DoD. The purpose of configuration management is to identify, control, account for, and audit the functional and physical characteristics of systems, equipments, and other designated material items developed, produced, operated, and supported by DoD components. Although the provisions of this regulation apply to all DoD components, it is not intended to impose strict unalterable rules on the implementing agency. Rather, the configuration management process is to be carefully tailored to the quantity, size, scope, stage of acquisition life cycle, nature, and complexity of the CI involved, whether the CI is developed at Government expense or privately developed and offered for Government use (AFR 65-3, 1975:1-1).

AFR 65-3 further states that:

> The selection of CIs to be configuration-managed is determined by the Governments' need to control a CI's inherent characteristics or to control that CI's interface with other items. The selection of prime and lower level CIs is basically a management decision normally accomplished through the system engineering process. The decision is based on numerous engineering/logistic factors (AFR 65-3, 1975:1-1).

However, configuration identification is applied to all hardware and computer programs. This identification is also the basis for the preparation of technical, administrative, and management documents. This documentation includes specifications, work breakdown structures, technical reports, test plans, test procedures, test reports, cost performance reports, and provisioning documents. In addition, configuration identification is the basis for configuration control and status accounting during the life cycle.

Configuration identification is defined as the process of documenting performance, qualification, fabrication, and acceptance requirements (AFSCP 800-7, 1977:16). The documented requirements are called a "baseline" which is implemented by multilateral agreements among the contractor, procuring agency, and user. Three common baselines (functional, allocated, and product) are described in the aggregate of specifications, drawings, parts lists, computer listings, and other documentation used to describe the functional and physical characteristics of a system. These baselines and their relationship to the system acquisition life cycle are depicted in Figure 5 (AFSCP 800-3, 1976:9-3). However, configuration identification is not necessarily synonymous with configuration baselines: unbaselined identification is used for visibility, and baselines are used for control. Identification constitutes a baseline only when formally designated and fixed at a specific point as a reference point for change control (AFSC Supplement 1, AFR 65-3, 1975:1).

In addition to the above definitions and guidelines, AFR 65-3 prescribes implementing instructions for Air Force organizations. The policies and procedures most relevant to this thesis are delineated below:

a. Computer programs will be managed as essential system elements using common procedures tailored to recognize their unique properties.

b. For each CI, an individual will be assigned the responsibility and delegated the authority for management of the item's configuration.

c. AFSC will select, in collaboration with AFLC or United States Air Force Security Service(USAFSS) and the Using command(s), the configuration items on which configuration status accounting will be initiated and maintained.

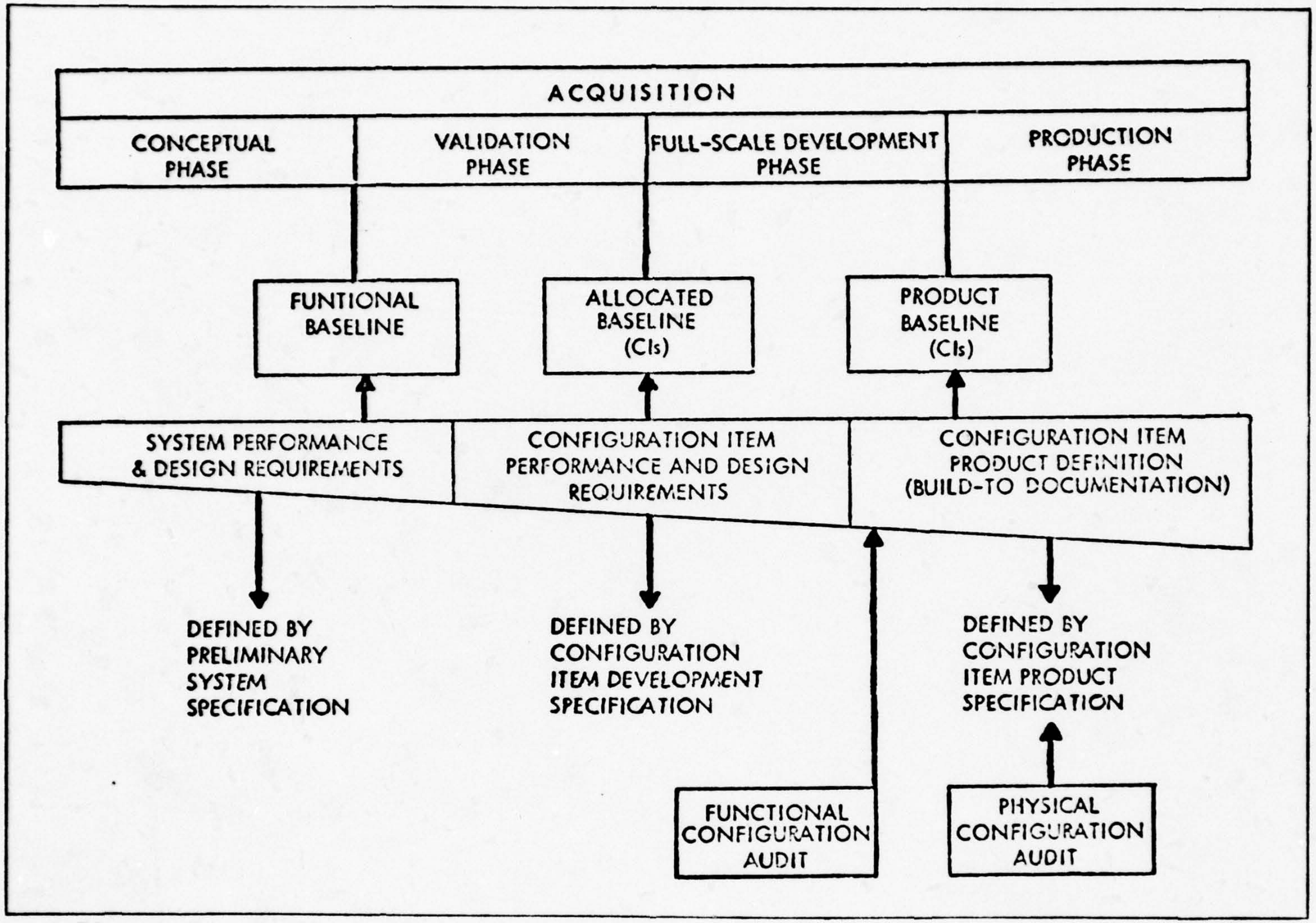

ACQUISITION
CONCEPTUAL PHASE
VALIDATION PHASE
FULL-SCALE DEVELOPMENT PHASE
PRODUCTION PHASE
FUNTIONAL BASELINE
ALLOCATED BASELINE (CIs)
PRODUCT BASELINE (CIs)
SYSTEM PERFORMANCE & DESIGN REQUIREMENTS
CONFIGURATION ITEM PERFORMANCE AND DESIGN REQUIREMENTS
CONFIGURATION ITEM PRODUCT DEFINITION (BUILD-TO DOCUMENTATION)
DEFINED BY PRELIMINARY SYSTEM SPECIFICATION
DEFINED BY CONFIGURATION ITEM DEVELOPMENT SPECIFICATION
DEFINED BY CONFIGURATION ITEM PRODUCT SPECIFICATION
FUNCTIONAL CONFIGURATION AUDIT
PHYSICAL CONFIGURATION AUDIT

Figure 5. Configuration Management by Baselines (AFSCP 800-3, 1976:9-3)

d. AFLC and USAFSS will, for material for which each is responsible, assist AFSC in the selection of configuration items to be managed and monitor their early development for logistics planning.

e. Configuration management will be applied to each CPCI throughout the system acquisition life cycle.

(AFR 65-3, 1975:F-1)

AFSCP 800-7. This pamphlet, entitled "Configuration Management," supersedes AFSC/AFLC Manual 375-7 and contains philosophy, methods, and procedures based on DoD and Air Force policy. It is intended to be a tutorial text as well as a reference for those who are working in configuration management daily. More specifically, the pamphlet states that:

> The actual application should be guided by AFR 65-3/AFSC Sup. 1 and the judgement and experience of the Program Manager and configuration management office (AFSCP 800-7, 1977:5).

Therefore, some flexibility is permitted in the decomposition of system requirements (defined in a system on system-segment specification) into lower level requirements that are subsequently specified in CPCI prime item development specifications. This flexibility is based on the idea that CIs usually cannot be definitely selected before final design of the system. In fact, this pamphlet states that:

> The initial CI list should contain more items than is anticipated on the final list. Deletion of an item is preferable to backtracking after production to build a data bank, since retrieval of data after the fact is difficult and costly (AFSCP 800-7, 1977:31).

Although the selection of CPCIs is not an exact science, it should be approached in the same manner as hardware CI selection. This leads to several criteria for use in selecting CPCIs.

This pamphlet specifies that while one major governing factor is that a CI should have characteristics that make it feasible to be produced by a single contractor and tested as an entity; another very important factor is the level at which there is a capability to manage. In other words, if too many CIs are established within a system, control may be lost. Therefore, in designing a system and selecting CIs, the management resources of Government program offices and contractors must be considered (AFSCP 800-7, 1977:9). In addition, separate CPCIs should be initially established for generic categories (e.g., operational, test, and support computer programs). A further division may be made for contract or subcontract purposes, for logistics management and operation, or to match up the system configuration with the statement of work and work breakdown structure (AFSCP 800-7, 1977:76).

These general guidelines are also supplemented with more definitive criteria for CPCI selection. They are based on the Air Force policy that CPCIs satisfy an end use function and are designated by the procuring activity as subject to configuration management procedures. Specifically, they are as follows:

a. Assign processes that interact strongly (e.g., in many or complex ways) to the same CPCI.

b. Assign processes with little or no interaction to different CPCIs.

c. Allocate to different CPCIs processes that will execute in different computers.

d. Assign to different CPCIs processes whose development can feasibly be finished at significantly different times, if such phased development will expedite overall system development.

e. Allocate to different CPCIs, computer programs to be separately procured.

f. Include in each CPCI no more than an individual Government monitor can efficiently track, assuming reasonable working relationships between him and the types of personnel who will manage and develop the CPCI.

(AFSCP 800-7, 1977:76)

In summary, the selection of CPCIs is one of the most critical decisions made during the acquisition of a major defense system. They provide the structure for technical direction, administrative actions, management decisions, visibility, documentation, and contractual delivery. Therefore, the criteria used for selecting CPCIs requires careful consideration by the contractor and System Program Office (SPO).

To emphasize the importance of this decision, and configuration management in general, the Statement Of Work(SOW) and contract for acquisition programs are tailored to the specific requirements of the system. AFSCP 800-7 also contains guidelines and instructions for applying configuration management contractually. Some of the instructions and sample clauses that have been used in Request for Proposals (RFPs) are provided in this pamphlet. Those relevant to this research project are:

a. Configuration identification should be established in the form of specifications and referenced technical documentation. This identification becomes more detailed as design and testing progresses.

b. The Allocated Configuration Identification(ACI) is documented by a performance oriented specification in accordance with MIL-STD-490, MIL-STD-483, and MIL-S-83490. This identification documents functional requirements allocated to a CI from a system or higher level CI and is contained in CI development specifications. The ACI is formally established during Advance Development/Validation or Full-Scale Development for the CIs.

AFSCP 800-3. AFSC Pamphlet 800-3 also establishes policies and implementing guidance governing the configuration management of systems, equipments, and computer programs. In most cases, the requirements are consistent with previously cited directives. However, this pamphlet, entitled "A Guide for Program Management," does perpetuate the confusion and conflict on whether software should be considered an item of data or an active system component. The guideline specifically states:

> For this guide and clarity, contractor prepared data acquired on DD Form 1423, Contract Data Requirements List, as a part of System/Equipment Procurements are essentially anything other than hardware as depicted by Figure 6 (AFSCP 800-3, 1976:16-2).

This policy is based on the requirements of the ASPR, Section IX and DPC Circular 74-3 which states that computer software falls within the definition of data. Although AFSC

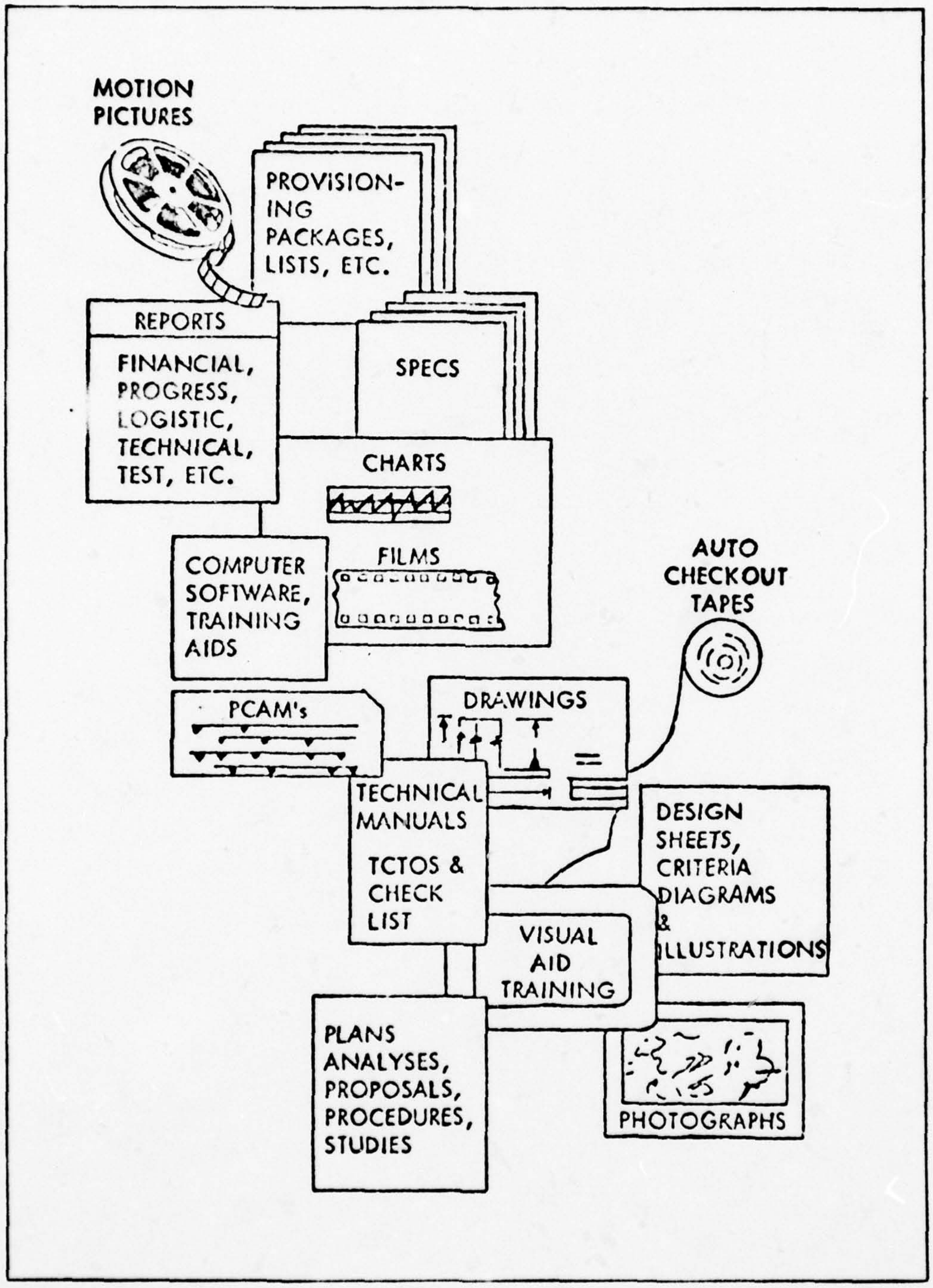

MOTION PICTURES
PROVISIONING PACKAGES, LISTS, ETC.
REPORTS
FINANCIAL, PROGRESS, LOGISTIC, TECHNICAL, TEST, ETC.
SPECS
CHARTS
FILMS
COMPUTER SOFTWARE, TRAINING AIDS
AUTO CHECKOUT TAPES
PCAM's
DRAWINGS
TECHNICAL MANUALS
TCTOS & CHECK LIST
DESIGN SHEETS, CRITERIA DIAGRAMS & ILLUSTRATIONS
VISUAL AID TRAINING
PLANS ANALYSES, PROPOSALS, PROCEDURES, STUDIES
PHOTOGRAPHS

Figure 6. Contractor Prepared Data (AFSCP 800-3, 1976:16-2)

has indicated that computer software acquisition should be subjected to the same contractual and cost controls as hardware (AFSCRP 800-1, 1975:3), confusion will abound until this directive and the ASPR are changed.

The selection of configuration items is also addressed in this pamphlet. Several factors are cited as the basis for making the selection decision. Those applicable to CPCI selection are:

a. Each system unit is designed to, and controlled by, one CI specification.

b. Each computer program is identified and documented by one top flow chart and a structure of subordinate flow charts and detailed descriptions.

c. However, in the final analysis, selecting CIs requires the exercise of certain judgements based on experience and program requirements. No set of rules can adequately describe this judgement factor.

d. While the selection of CIs on a program is usually made through the cumulative recommendations from several sources, it is also an area where early decisions can benefit the program. This is not to say that CIs can be definitely selected before final design of the system, for flexibility is a necessity. It would be desirous for a preliminary list to be compiled as early as possible.

e. It is very important to include the AFLC System Manager(SM) in the process. With the advent of new weapon systems, it is expected that they will be selected by AFLC for application of the AFLC Advanced Configuration Management System. In most cases, SM selections will parallel those made by the SPO and again, cost savings can be realized if these are resolved in the early program stages.

(AFSCP 800-3, 1976:9-4)

Military Standards

Military Standards covering configuration management requirements and procedural guidance are the basic documents for contractually implementing a configuration management program on acquisition projects. They provide guidelines to both government and contractor personnel responsible for one of the various aspects of configuration management. One of these, MIL-STD-483 (USAF), entitled "Configuration Management Practices for Systems, Equipment, Munitions, and Computer Programs," establishes uniform configuration management practices that can be tailored to all USAF systems. In other words, contracts invoking this standard may identify the appropriate paragraphs and appendices as applicable to that acquisition project.

Appendix XVII of this standard specifies that the selection of CIs is based on the definition contained in DoDD 5000.19, i.e., a CI is "an aggregation of hardware/software, or any of its discrete portions, which satisfies an end use function ..." Furthermore, CI selection reflects an optimum management level during acquisition. This level is one at which the procuring activity specifies, contracts for, and accepts individual elements of a system (MIL-STD-483, Notice 2, 1979:129).

Several general and specific considerations are presented in this recently published appendix, entitled "A Guide for Selecting Configuration Items." Most of the general guidelines are covered in previously summarized directives. However, the following ones deserve specific mention:

> a. Selecting CIs should be with a full view of the life cycle cost and management impacts associated with such a designation. Choosing too many CIs increases the cost of control; choosing too few or the wrong elements as CIs runs the risk of too little control through lack of management visibility.
>
> b. The major elements comprising the system should be identified as CIs during the Demonstration/Validation Phase. Early selection of CIs is important since management emphasis becomes greater as development progresses. As development continues and logistic and technical considerations surface, additional items can be designated as CIs. Usually, the CI selection process should be essentially complete by the Preliminary Design Review.
>
> (MIL-STD-483, Notice 2, 1979:129-130)

Some of the specific considerations upon which the selection decision shall be based are:

> a. This selection is usually a technically driven decision made by the developer. It is based upon system trade-offs and the natural decomposition of the software. Premature partitioning must be avoided because, to a certain extent, it may preordain the design.
>
> b. Generally, the executive/supervisor, functional/applications, input/output, test, and support programs should be individual CPCIs. Computer programs with potential use in multiple systems should be separate CPCIs. Highly interrelated computer programs should be combined as one CPCI. CPCIs should be established to their largest functional element (i.e., Operational Flight Program, Flight Simulator Computer Program, etc.).
>
> c. Assemblies of computer program elements to be acquired from a single contractor are potentially a single CPCI.

d. Computer programs to be designated for operation in different models of computers should be separate CPCIs.

e. Computer programs scheduled for development, testing, and delivery at different times may be separate CPCIs.

(MIL-STD-483, Notice 2, 1979:131-132)

As previously indicated, the effects of improper CPCI selection can be devastating to an acquisition program. All three evaluation parameters (cost, schedule, and performance) for the government, prime contractors, sub-contractors, and suppliers can be adversely affected. A detailed listing of the usual effects of CI desigantion is included in this appendix to MIL-STD-483. In addition, a series of questions which makes up a checklist to be used in selecting CIs is provided. According to this standard, the system component probably should not be a CI if most of the questions can be answered "no." Likewise, if most of the questions can be answered "yes," the item should be a CI. Additional judgement is needed if the questions can be answered with approximately equal numbers of "yes" and "no." Since this thesis concerns the criteria for CPCI selection, the checklist is reproduced below:

a. Is it a critical high risk, and/or a safety item?

b. Is it readily identifiable with respect to size, shape, and weight (hardware)?

c. Is it newly developed?

d. Does it incorporate technologies?

e. Does it have an interface with hardware or software developed under another contract?

f. With respect to form, fit, or function, does it interface with other items whose configuration is controlled by other entities?

g. Is there a requirement to know the exact configuration and status of changes to it during its life cycle?

(MIL-STD-483, Notice 2, 1979:134)

Chapter Summary

The acquisition and management of computer resources within the Federal Government is controlled by a complex hierarchy of policies and directives. In general, those applicable to the acquisition and management of computer resources during the development, acquisition, deployment, and support of major defense systems were summarized. More specifically, the DoD and AF policies, procedures, and guidelines for the allocation of system software requirements to CPCIs on a large software-intensive system were emphasized. Unfortunately, only a few of the documents address the selection of CPCIs directly. On the other hand, several do present guidelines and considerations for applying configuration management concepts to the acquisition and management of software.

At the top of the regulatory structure governing the acquisition and management of computer resources is Public Law 89-306. This law, called the "Brooks Bill," provides the basic structure and concepts for the government-wide system of ADPE management. OMB Circular 74-5 specifically excluded those computer resources integral to weapon systems and specially designed computer equipment from control of the centralized procurement concept of P.L. 89-306. This exception was implemented via OMB Circular A-109, DoD Directives 5000.1, 5000.2, and 5000.29, the Air Force 800-series regulations and the Air Force Systems Command 800-series pamphlets.

In essence, these documents

a. define "major system acquisitions" and provide the policies, procedures, and guidelines for their management,

b. establish policy for the management and control of computer resources that are part of a major system acquisition,

c. require that computer resources be managed as elements or subsystems of major importance during the system acquisition life cycle, and

d. specify that both computer hardware and software will be treated as configuration items.

The requirement for applying configuration management concepts to software led to the development of several other policies and regulations. Those applicable to the selection of CPCIs are DoD Directive 5000.19, DoD Instruction 5010.21, AF Regulation 65-3, AFSC Pamphlet 800-7, and Military Standard 483. Although these documents in general establish uniform policies and procedures for the implementation of configuration management within the DoD, the actual application to software is often a management decision which requires the judgement and experience of the Program Manager. For example, the selection of CPCIs is not an exact science. As a result, several criteria for use in allocating software requirements to CPCIs are presented in these documents. In summary, they are as follows:

a. The initial CI list should contain more items than is anticipated on the final list.

b. The CI should be produced and tested by a single contractor as an entity.

c. Assign processes that interact strongly to the same CPCI.

d. Allocate to different CPCIs processes that will execute in different computers.

e. Assign to different CPCIs processes whose development can feasibly be finished at significantly different times.

f. Include in each CPCI no more than an individual Government monitor can efficiently track.

g. Selecting CIs should be with a full view of the life cycle cost and management impacts associated with such a designation. Choosing too few or too many CIs adversely affects the program.

h. The selection of CPCIs is usually a technically driven decision made by the developer as a result of system trade-offs and the natural decomposition of the software.

One other significant finding of this review was the ASPR treatment of computer programs. Considering software as an item of data in the same manner as reports, forms, manuals, and specifications causes confusion and conflict. AFSC has initiated an attempt to have the ASPR committee reverse its decision. In the meantime, AFSC has continuously required that computer software be subjected to the same contractual and cost controls as hardware.

IV. Non-Regulatory Publications

Over the past few years, there has been a steadily sharpening focus on what has been characterized as the "DoD software problem." This focus is evidenced by the large number of non-regulatory publications addressing software acquisition and management problems. This has stimulated senior decision makers, most of whom have no personal experience with software, to learn about it and give it management attention commernsurate with the $3 billion DoD spends on it every year (Carlson, 1976:379). As a result, the DoD and its components have sponsored, or participated in, numerous conferences, workshops, and studies for the purpose of developing a better understanding of the so-called "software problem." In addition, a large number of technical reports and management guidelines have been either prepared or supported by the DoD and its agencies. Despite these efforts, some weapon system acquisition programs are still experiencing software cost growths and overruns, schedule delays, and unreliable performance.

There is a considerable body of reputable sources available which document these undesirable consequences. It would not serve the purpose of this thesis to summarize each one for two reasons. First, the number of studies, reports, and lessons learned is overwhelming. Secondly, the specific topic of this research is the process of allocating software requirements to computer program configuration items in the acquisition and management of a large software-intensive weapon system. Therefore, only those studies, reports, and lessons learned which are relevant to this subject are considered in this chapter.

Other sources reviewed in this literature search include conference proceedings, textbooks, periodicals, and management guidebooks. In most cases, these sources are directed at specific functional areas, as opposed to the problems associated with CPCI selection in the acquisition and management of computer resources. Some are aimed at identifying problems only, while others address both problems and solutions. In addition, a large amount of "expert" opinion, as well as factual material, is contained in these sources. Therefore, only those that address the process of CPCI selection and its impact on the overall program are summarized below.

DoD Sponsored Studies

As early as 1964, high-level Air Force management had already recognized that computer technology was developing rapidly and that expanding requirements would dictate a proliferation of applications. It was predicted (accurately) at that time that the Air Force computer inventory would increase by 50 percent in the ensuing 2 or 3 years. That prediction was followed by the awarding of a contract to the Planning Research Corporation for a "Study of Application

Effectiveness and Problems of Air Force Information Processing Systems." Although several other studies addressed problems with the computer system acquisition process in the ensuing 11 years, few of the recommendations were ever implemented. In summary, the various studies had some minimal effect; however, they failed to achieve the positive action required to solve the problems (Drezner, et al., 1976:3).

The Computer Resources Management Study. In early 1975, the Air Force requested that The Rand Corporation examine AF management of computer resources. Basically, the study considered the broad aspects of management, policy, and organization relevant to the computer resources for major weapon, command and control, telecommunication, intelligence, and management and support systems. The purpose of this study was not only to identify problems but to propose alternative courses of action in policy, management, and organization and to identify the important consequences of implementation of the alternatives. The only finding/recommendation directly related to the requirements allocation process was that "the requirements process is exceedingly important." Furthermore, "an inadequate focus and emphasis is placed on the requirements process" (Drezner, Shulman, and Ware, 1975:5-8).

The MUDD Report: A Case Study of Navy Software Development Practices. This report is the result of a year-long investigation into Navy software problems. The purpose was to describe some of the problems encountered in producing reliable software in a context familiar to Navy software developers in the hope that they could recognize and avoid similar errors in the future. A mythical software system was used to describe where and how the developers went awry. The pitfalls outlined typify problems which actually occurred in software developmental efforts. One of the problems chronicled in this report was stated as the most important design question confronting the system designers. That is, "how to partition the system in such a way that it would be easy to modify." Furthermore, the author indicates that project managers showed a lack of awareness of the key issues involved in partitioning software when questioned on development strategy (Weiss, 1975:16).

DoD Weapon Systems Software Acquisition and Management Study. In December 1974, the DoD initiated a two-phased software acquisition study program to identify methods for controlling increasing costs, improving the quality, and minimizing the adverse impact of software in weapon systems. The MITRE Corporation and the Applied Physics Laboratory of Johns Hopkins University agreed to perform separate, but coordinated, four-month studies in support of the study program. In Volume I of MITRE's report, entitled "Findings and Recommendations," the MITRE Corporation identified several major systems software problem areas facing the DoD, their causative factors, and the high payoff areas that should be given priority consideration by management. In the area of software management methods, the report stated that many of the software acquisition and management problems can be traced to inadequate requirements formulation and the need for more detailed planning during the early stages of weapon system acquisition (Asch, Kelliher, Locher, and Connors, 1975 a:2-7).

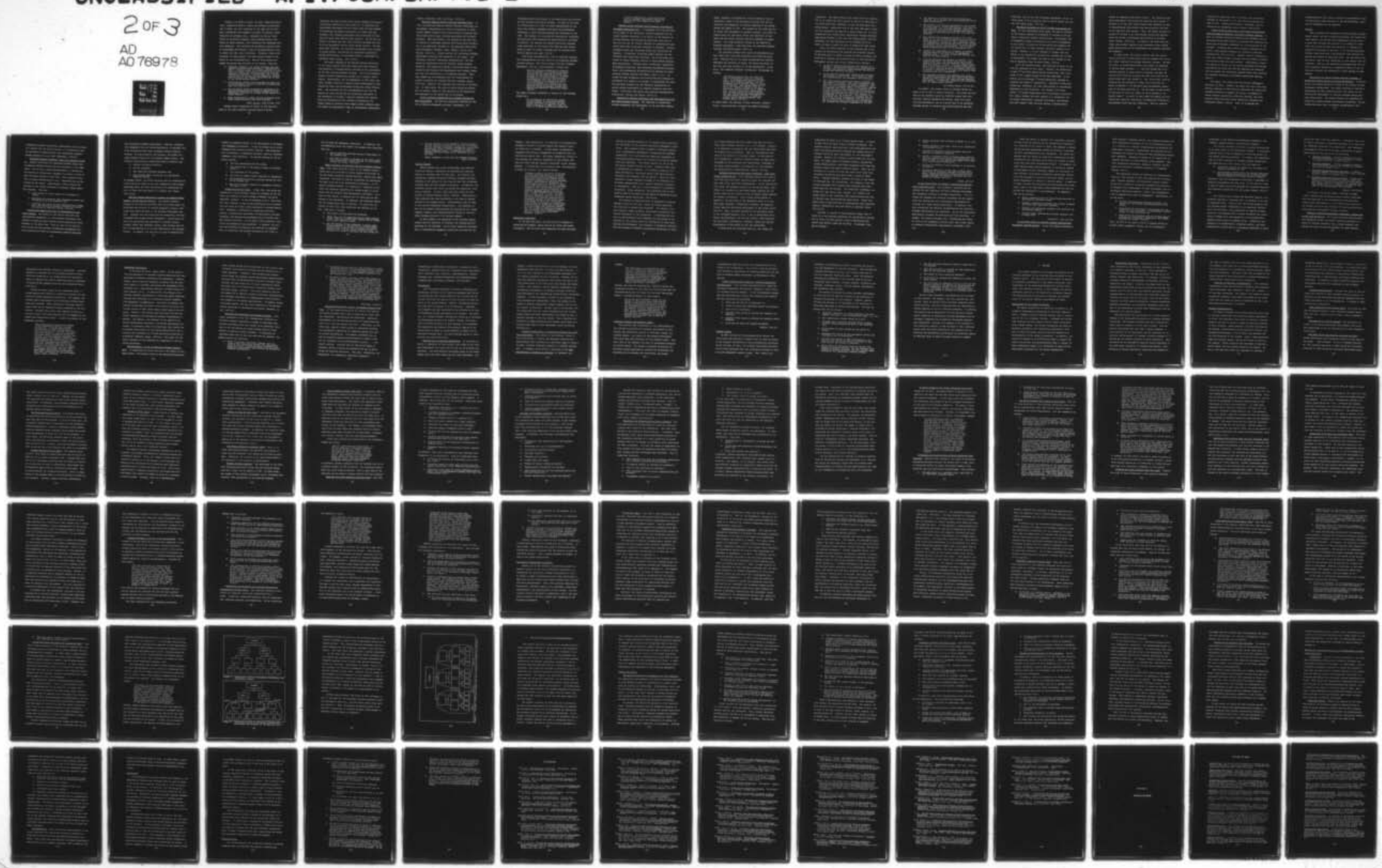
To recommend a course of action for improving the management and control over costs, the quality, and the timeliness of software in weapon systems, MITRE extracted from the study findings the areas of highest payoff. One of those areas felt to have the greatest leverage is the Software Performance Specification. The corrective actions in this area concern the recognition and consistent application of sound engineering principles and practices to the process of specifying software end products, i.e., CPCIs (Asch, et al., 1975 a:2-13). A checklist of important factors to be applied across all systems was defined and recommended to be the subject of required DoD guidelines. Those relevant to this research are as follows:

> a. Choose a software architecture which best reflects the weapon system requirements.
>
> b. Emphasize ease of change in the software performance specification process. Recognize that weapon systems software requirements will change over the life of the system. Where appropriate, consider use of a modular architecture which allows for changing application program requirements.
>
> (Asch, et al., 1975 a:3-7)

AD-A076 978 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOO--ETC F/G 15/5
AN INVESTIGATION OF THE SOFTWARE REQUIREMENTS ALLOCATION PROCES--ETC(U)
SEP 79 V L COOPER
UNCLASSIFIED AFIT/GSM/SM/79S-2 NL

2 OF 3

AD
A076978



Volume II of MITRE's report, entitled "Supporting Material," summarized several studies and workshop proceedings. One of those, the 1973 Symposium on the High Cost of Software, considered what research is needed to achieve a major reduction in software costs. Sponsored by the Air Force Office of Scientific Research, Army Research Office, and Office of Naval Research, this conference was organized into five workshops. Each workshop was assigned a specific area to investigate and develop a list of recommendations on how to reduce the high cost of software. Workshop 5 dealt with "problems of large systems" and made several points in addition to its recommendations. Four of these points concerned the design and structure of the system. They are:

a. Premature "freezing" of system structure into management structure causes problems. (Researcher's Note: In other words, it is very difficult to overcome the inertia that naturally develops once the management structure is established. As a result, efforts are duplicated, system integration becomes more difficult, and overall visibility may be lost as the system design evolves.)

b. Configuration control can be applied too early and at too detailed a level to allow for continuing design that must go on.

c. The procedures usually followed in government contracted software are not flexible enough to allow significant tradeoffs to be made during design in a timely fashion.

d. Tight configuration control should be delayed until major design questions have been answered.

(Asch, et al., 1975 b:3-29, 3-31)

Another report summarized in Volume II was that prepared by the Army Scientific Advisory Panel's Ad Hoc

Committee for Army Tactical Data System Software Development. The primary objective of this group was "to determine the exploratory development efforts which offer the best promise of satisfying the Army's requirement to reduce cost and schedule, and increase performance and reliability of software for major Army tactical data systems." One of the committee's findings was that "many software problems are rooted in an inadequate initial system design effort." To mitigate these problems, the committee recommended that an orderly system design, considering all reasonable alternative system architectures, be performed before a development is initiated (Asch, et al., 1975 b:3-60).

The report prepared by the Applied Physics Laboratory of the Johns Hopkins University summarized their findings and presented 17 recommendations directed at the alleviation of the more serious software problems. One of the recommendations concerns identifying software as a contract deliverable. Specifically, the recommendation is that major computer software involved in weapon systems development be designated as CPCIs and deliverables during Full Scale Development. The rationale for this recommendation was that the absence of clear definitions and guidelines applicable to the components of software (computer programs and computer data) as distinguished from software documentation has caused numerous instances where weapon system contracts have not called for appropriate items as deliverables (Kossiakoff, Sleight, Prettyman, Park, and Hazan, 1975:2-5).

Ballistic Missile Defense Reliable Software Study. In 1975, the Ballistic Missile Defense Advanced Technology Center(BMDATC) commissioned the LOGICON Corporation to study current BMDATC research efforts and develop a research plan that would explore basic concepts associated with reliable software. More specifically, the purpose of the study was to develop an integrated approach that would allow reliability to be specified, designed in, and measured during software development (Ikezawa, Kundu, Lambert, McGill, and Nielsen, 1975:1-2). Several methods for performing software requirements analysis were presented in the report. One technique, entitled "Functional Decomposition," was discussed in conjunction with the top-down design process. Basically, this approach produces a functional model of the system by partitioning the requirements into threads, where each one is a description of a subsystem function. Then, each thread can be decomposed into the tasks/modules required to perform the intended function (Ikezawa, et al., 1975:3-5). However, this treatment was strictly a technical one. In other words, the level at which CPCIs are defined and the overall impact of that decision on the acquisition and management of the system was not discussed.

Joint Logistics Commanders(JLC) Software Reliability Work Group(SRWG). The SRWG was originally chartered by the JLC to develop recommended policies, procedures, and techniques which would result in the acquisition and fielding of more reliable electronics systems. In order to overcome the difficulties of a lack of common understanding of the basis for software management practices and engineering technology, a multi-faceted approach was undertaken by the SRWG. For example, all existing DoD regulations, standards, and directives were examined for applicability to software. In addition, all ongoing and planned technology programs being undertaken by the various DoD agencies were reevaluated in terms of current problems and low software reliability (Manley and Lipow, 1975:3).

From these investigations, a set of detailed findings and recommendations for the solution of recognized software problems emerged. The first finding, entitled "Software Management, Policies, and Procedures," stated that

> Existing policies governing the acquisition of software for military electronic systems are not adequate to ensure the delivery of reliable software. Although software is now being established as a configuration item, the lack of sufficient emphasis in existing policies on the early design of software as a CI tends to generate reliability and other problems during subsequent systems production and operation (Manley and Lipow, 1975:7).

The impact statement presented in support of this finding stated that

> The reliability of operational weapon system software is seriously degraded because treatment of software as data rather than as a CI leads to a lack of focus on software reliability issues and

> a lack of appropriate visibility and review of software development during the acquisition phase (Manley and Lipow, 1975:7).

Defense System Software FY79-83 Research and Development(R&D) Technology Plan. In September 1977, the R&D Technology Panel to the Management Steering Committee for Embedded Computer Resources in the Office of the Director of Defense Research and Engineering formulated this plan to help the DoD and the industry managers in planning and evaluating their R&D initiatives relating to software. Several software science and technology areas were identified for research. In addition, projected budget profiles for the FY78-FY83 time frame were specified. One of the areas projected to reveal promising results is Requirements Analysis. More specifically, "Software First" technology, competitive prototyping guidelines, semi-automated requirements analysis tools, and requirements decomposition are planned for basic research thrusts (Carlson and DeRoze, 1977:i, I-1). Although the plan recognizes that conflicting and changing requirements will always be a problem, it establishes the need for providing insight into the technical implications of stated system requirements on computer resources (and vice versa), identifying risk areas, and exploring implementation alternatives (Carlson and DeRoze, 1977:I-1).

Operational Software Management and Development for U.S. Air Force Computer Systems. The Committee on Operational Software Management and Development of the Air Force Studies

Board, Assembly of Engineering, National Research Council, conducted a study of the problems associated with the acquisition and management of software employed in AF systems during the summer of 1976. The purpose of the study was not to review USAF management of specific systems, but rather to examine ways the USAF might become more effective in managing and developing computer resources embedded in weapon systems. The study primarily focused on two aspects of software management: risk reduction, and cost and schedule problems (AF Studies Board, 1977:1-3).

The committee offered seven major recommendations and many other minor ones that reinforce and extend the major ones. Although none of the major recommendations directly addresses the selection of CPCIs, the second one would impact the selection decision if it was implemented. In essence, the committee endorsed "incremental" development by stating,

> "Experience teaches that too often too much is attempted at once in a development. To mitigate this problem, we recommend that wherever possible, the development should proceed through a series of deliveries. Each should stand by itself, not require a rework of its predecessors, and each should be of tangible use. During this period, strong feedback is needed from the using organization and the maintaining organization to the development activity, in order to correct any recognized problems" (AF Studies Board, 1977:10).

In other words, the delivery of each increment, called a "block," would provide a complete and useful functional capability. Two major reasons were stated for this approach. One is to reduce the total amount of code to be developed in the first block and, therefore, to get the core parts operating sooner. Along with this is the fact that such a strategy will reinforce the need for a sound design for the total program and will better force independence between different application modules in the code (AF Studies Board, 1977:36). Therefore, the CPCI structure of the software could be based, at least in part, on what is planned for each block.

Since the fifth objective of this research was to determine the feasibility and evaluate the potential impacts of defining CPCIs in terms of system versions or models, the committee's findings and recommendations on the incremental development concept are briefly summarized as follows:

a. It may be argued this approach will stretch out a project. The committee thinks not, especially not in high risk projects, where problems may prevent almost any effective dclivery.

b. It may also be argued that breaking some programs into blocks is artificial. If it truly is, then the committee does not recommend this technique.

c. The block concept should also permit tradeoffs in the content of blocks. This will accommodate problems while maintaining the overall schedule. However, to guard against casual or arbitrary slipping of features into later blocks, an incentive fee contract structure that rewards the contractor for on-time delivery by block and function, and denies rewards when features or blocks are delayed is recommended by the committee. The incentive fee is a major tool for keeping the schedule and content in each block in agreement with the contract. But it is still only a tool and not a substitute for informed and diligent management by the SPO.

d. The approach is compatible with DoD Directive 5000.29, but it appears that USAF Regulation 800-14 needs modification.

e. The concept is of general applicability and should be first used in a high risk project. If, then, it is successful, the committee believes review will show that the success is for reasons which have more general applicability than just for high risk projects.

f. Each operational program should be a CPCI, with block I, II, III deliveries adding up to one of the CPCIs called for in the contract. Software tools for development, operations, and maintenance should be separate CIs with deliverables for users and for maintainers clearly distinguished. But, resident debug programs are part of the operational CPCI.

g. "Blocks" are subdivided into "builds" or similar substructures and "builds" are the management tracking items. The flow of work and reviews proceed in the usual fashion.

h. A system developed and managed by "builds," or a similar structure, will focus attention on user capabilities which are demonstrable, distinguishable and separate in code. This will allow "acceptance" on a block basis, with block documentation. As a result, manpower peaking is reduced, structure in design is emphasized, and early and orderly user evaluation and use is permitted.

i. The committee believes this approach will reduce the risk in development and will yield lower development and life cycle costs and a shorter development period. In addition, surprises and scrambling will be significantly reduced.

(AF Studies Board, 1977:31-54)

In summary, the current trend in software design appears to be to give it better structure. This makes development of the different parts by different people much more feasible, permits staggered deliveries, makes for better test and maintenance, and is a method much to be preferred. Top-down design is currently well-publicized and has been effective, but is only one of several approaches, no one of which seems to be so superior that it should become the only way (AF Studies Board, 1977:54).

The Evolving Nature of the C^3 Systems Acquisition Process. In this unpublished white paper, the author reflects on the lessons learned in developing C^3 systems at the Air Force's Electronic Systems Division(ESD). He proposes that a fundamental change in attitude on the part of management at all levels in regard to the procedures and emphasis for the acquisition of these software dependent systems is required. However, it is not intended to suggest that the fabric of our acquisition policies are outmoded and in need of replacement, but rather the need is for a change in the way we implement those policies (Doane, 1979:1).

As in the previous study, the selection of CPCIs is not directly addressed in this paper. However, the differences between C^3 and other weapon system acquisitions, and a sampling of the lessons learned by ESD are described. In addition, an evolutionary C^3 system acquisition strategy is proposed for future C^3 acquisitions. Embodied in these presentations, therefore, are some implications of considerable importance to CPCI selection. For example, the author states, "C^3 system requirements are intrinsically evolutionary, partly because they must operate in a constantly, but not always predictably, changing environment, and because they must support human decision making, a process which cannot be completely specified a priori. The prevailing Air Force development and acquisition practice which treats C^3 systems as if they were stable rather than changing, as akin to mass-produced items rather than one-of-a-kind, has been out of step with this reality. Thus, the author proposes to subject C^3 systems to an evolutionary acquisition approach where the system is developed and delivered in multiple steps" (Doane, 1979:9-13). In essence, the first step provides some minimum capability and subsequent steps contain added capabilities as the requirements are defined, approved, and budgeted.

In other words, the author believes that the overall thrust of future C^3 acquisitions must be changed from building "ideal" systems to building "workable" systems. To make his point, he quotes Watson Watt, who stated (in reference to British radar systems), "... The best design had to be rejected because it would never be achieved, and that the 'second best' would be achieved too late to be used by the armed forces when they needed it. The 'third best' would be adequate and was available in time, and it was what won the Battle of Britain." He related a more contemporary expression of the same principle as: "If war comes, a good system in the field is worth any number of ideal systems on the drawing board." In short, we must lower our sights and work toward more realistic objectives, incrementally building as requirements evolve and are understood. That is, quantum increases in capability must be avoided, and incremental transfer of the system to the user must be accepted as a fact of life. System architectures, therefore, must be structured to accommodate change (Doane, 1979:14-18).

Joint Logistics Commanders Joint Policy Coordinating Group on Computer Resource Management-Software Workshop. A panel on Software Acquisition/Development Standards was established during this workshop to evaluate the potential for developing tri-service standards for the acquisition of ECS software. Currently each service acquires software for ECS within a very general framework as typified by Military Standards 483, 490, and 1521. However, specific implementations vary widely between services and even between contracting agencies within the same service. Additionally, experience gained in the use of the initial standards for software acquisition has shown the need for updating and expanding them to account for experience, earlier errors and omissions, and improved techniques of software engineering (Munson, 1979:1).

Once again, this report did not directly address the selection of CPCIs. However, the panel did look at how each service responded to DoDD 5000.29, evaluated the current Mil Standard structure to support the policy, hypothesized a common software acquisition life-cycle process, and where possible identified areas where current Mil Standards are inadequate (Munson, 1979:2). Thus, it is through the implementation of this panel's specific recommendations that the previously cited conflicts in the DoD Directives, ASPRs, and Mil Standards will be resolved.

Reports

Many government and industry technical reports, masters degree theses, Air War College publications, Air Command and Staff papers, and Defense System Management College study reports were reviewed and analyzed in an attempt to understand the various policies and positions on software requirements allocation in the DoD. As in the DoD sponsored studies, there were only a few documents that directly addressed the process of selecting CPCIs. For the sake of organization, they are all summarized in this section with one exception. That is, the guidebooks developed under the Air Force Guidebook Program as a cooperative effort of the Air Force and industry are reviewed in a later section of this chapter.

Development of Weapon Systems Computer Programs: A Method for Control During Full-Scale Development. Presented during the Program Management Course at the Defense Systems Management College(DSMC), this paper attempted to provide a methodology and some guidelines which will assist in controlling the development of computer programs during full-scale development. The author stresses the importance of proper and complete system requirements allocation. He concluded that since this is normally accomplished by the contractor's systems engineering organization and the result is a subject for the design reviews, it is imperative that the government Program Manager have experienced computer systems personnel on his staff (Zabriskie, 1975:20).

Reliable Computer Software: What It Is and How to Get It. Several techniques for improving software design and increasing software reliability are presented in this DSMC paper. One of them, called modularization, is defined as the process of decomposing a program into parts in order to make the system easier to deal with. Many important benefits of proper modularization (e.g., manageability) are advocated by the author. Additionally, three criteria for dividing up the effort involved in building a large system are proposed. They are:

a. tasks should be small enough for one person to handle,

b. decompose the system so that management control and flexibility can be maintained, and

c. partition the system so that implementation, debugging, and testing of modules or groups of modules can proceed in parallel (Farnan, 1975:19).

Configuration Management for the Development of Computer Systems. The author of this report concluded that configuration management is a well-defined discipline when applied to classical major defense system acquisitions where hardware is the prime item. That is, most of the publications within the DoD consider configuration management as a vital function and there is considerable guidance available for controlling hardware development. However, configuration management has had limited application to embedded computer systems and even less to general purpose computer systems. To help alleviate this problem, the author proposes several actions for the Systems Program Office. For example, CPCIs should be identified based on whether they

a. satisfy an end-use function,

b. are key programs,

c. have expected interface problems, and

d. will provide added visibility to a particular computer program.

In summary though, the author believes that the selection of a CPCI should be based on the best judgement of the program management team, and this flexibility will undoubtedly result in CPCIs that vary greatly in program size (Anway, 1976:13-23).

<u>Applying Program Management Concepts to Software Development: An AFR 300-15 Critique</u>. This report proposes changes to the recently published AFR 300-15 which adapt the principles and concepts of program management (phases, baselines, and formal reviews) to the software development process. Although both the report and the regulation, entitled "Automated Data System Project Management," specifically address the acquisition and management of general purpose systems, rather than embedded systems, some factors that affect the designation of a CPCI are described in the proposed changes. In summary, the term CPCI is used to describe the

product or products subject to the application of configuration management procedures. It may be defined as an entire automated data system or portion thereof. That determination is basically a management decision, based on tradeoffs, judgement, and experience. The factors bearing on the decision include:

a. the risk associated with the system,

b. the planned use of the system (single or multiple users),

c. the complexity of the system,

d. the level of change control required by management,

e. the modification activity expected during the life of the system, and

f. the cost thresholds imposed by management (Caudill, 1977:Atch 1, 21).

System Acquisition Guide. In May 1978, this guide was prepared as a group research project by students of the Air Force Air Command and Staff College at Maxwell Air Force Base(AFB), Alabama. It was not designed to be a "how-to-do-it" book; instead, it was written principally as an orientation for those with very little experience in system acquisition. Therefore, information is summarized from many sources, including DoD Directives, AF Regulations, Military Standards, and manuals and guidebooks from AF Systems Command Product Divisions. In the area of CPCI selection, very little new information was presented. In essence, the process was described as requiring the exercise of judgement based on experience and program requirements for which no set of rules can adequately compensate. In addition, two considerations which can benefit the program were presented. They are:

a. the CI selection decision should be made early in the program, and

b. care must be taken to include the Air Force Logistics Command(AFLC) system manager in this process (Runkle and Smith, 1978:106).

Model Statement of Work(SOW) Task for Software Development. This paper presents a model of the general requirements task in a SOW used by System Program Offices(SPO) at the Air Force Electronic Systems Division, Hanscom AFB, Massachusetts. The model is used as a guideline for specifying the work to be bid on by prospective contractors. In the area of software design, the contractor is required to document the rationale supporting his system design in accordance with the Contract Data Requirements List(CDRL). As a part of that design, a complete list of CPCIs must be presented to the Government. In addition, the rationale behind the CPCI organization must be defined. To assist the contractor in this task, four guidelines are provided by the SPO. They are as follows:

a. The number of CPCIs shall be minimized.

b. CPCIs shall not be organized across vendor product lines. That is, a CPCI shall operate and be tested as a single computer system.

c. For the purpose of this paragraph, a unique operating system(OS) is defined to be a single vendor product which controls the allocation of computer resources for a single vendor computer system. A separate CPCI shall be provided for each unique OS.

d. For the purpose of this paragraph, unique software utility services are defined as the set of software utility services (compilers, assemblers, diagnostics, and editors) which a vendor provides to support a single computer system. A separate CPCI shall be provided for each unique set of software utility services.

(Model Statement of Work Task for Software Development, 1979:4-5)

Lessons Learned

While software has become increasingly more sophisticated and complex, techniques for producing and ensuring quality in software have not kept pace. As a result, many acquisition programs encounter difficult software problems. Overruns of 100 percent in both cost and the time to develop software have not been unusual occurrences. In addition, the delivered system often fails to provide the capability originally required. In fact, there have been cases of total failure to develop systems (Davis, 1978:19). Traditionally, these problems have been documented in the form of lessons learned by the program office. Unfortunately, most of them do not receive widespread dissemination. One exception is the set of lessons learned during the acquisition of a large command, control, and communication(C^3) system by the Electronic Systems Division of AF Systems Command.

In that program, the magnitude of the problems spawned several serious inquiries to determine the program-specific solutions to the problems. One of those inquiries concluded that a configuration management system did not exist on the program. More specifically, the functions of configuration management existed in varying degrees and numerous forms, but there was no standard. In some cases, configuration management had become so fragmented and dispersed that its identity was almost lost. The lesson learned as a result of this problem was elegantly summarized in the October 1978 issue of "AFSC Computer Resource Newsletter" by Mr. Charles Bobbish. In supporting his conclusions that "the selection of CPCIs is critical," Mr. Bobbish stated,

> "Software acquisition should be managed in terms of system models or versions, each of which performs end-use system functional capabilities, and is an evolutionary outgrowth of previous versions. Further, configuration management of these models should be limited to baselining and controlling (in the allocated baseline), the system capabilities contained in each version as opposed to baseline detailed design information. Why? If one chooses a unit of management (in this case, a CPCI), the end result of managing that unit should be the ability to directly observe that a meaningful portion of the system works. A structure in which the "functional flow" across the system is divided into several CPCIs is artificial; management of a CPCI under this structure forces the SPO to make an inductive evaluation that the system will ultimately work if and only if all other CPCIs work" (Bobbish, 1978:3).

Management Guidebooks

In the past few years, the discipline of software engineering has emerged, taking form as an entity and formal discipline. The Air Force has recognized this and initiated efforts to provide guidance to its program office personnel. The key effort is the Software Acquisition Management Guidebooks. Each guidebook takes a specific functional subject (e.g., Configuration Management, Contracting for Software Acquisition, etc.) and emphasizes practical approaches to real-world problems within the context of DoD and AF regulations and standards. This effort started in 1975 at the Electronic Systems Division of AFSC. Currently three sets of guidebooks are underway: C^3, Airborne (Avionics, and Space and Missile), and Ground Based (Crew Trainer/Flight Simulator, and Automatic Test Equipment). Three sets were required to insure adequate coverage of the different application perspectives and to capture different viewpoints of preparation. Each set is being prepared by a different contractor. After completion, the guidebooks will be analyzed to see if the texts can and should be published as one integrated series (Marciniak, 1978:34-35).

The process of selecting CPCIs is addressed in varying degrees in each of the sets. Basic principles governing the selection in general are presented. For example, lessons learned, common pitfalls, and mistaken assumptions are typical treatments in the guidebooks. A comprehensive summary of each guidebook that addresses CPCI selection is not provided in this thesis since the information is often overlapping and furthermore, this research is primarily concerned with the process of software requirements allocation to CPCIs on a large software-intensive system that does not inherently have a natural breakout (e.g., a C^3 system). However, it is important to emphasize the one common theme expressed in several of the guidebooks. That is, the selection process is not subject to "stylized" rules. In the words of one C^3 guidebook, "Decisions should be based on experience, knowledge of the principles and implications, knowledge of the given system program, and attention to both technical and administrative considerations" (Searle, 1977:21).

Software Acquisition Management Guidebook: Life Cycle Events (ESD-TR-77-22). This guidebook explains the acquisition life cycle for major defense systems, the computer program life cycle, and their relationships. In the acquisition life cycle, the primary objective of the second phase, validation, is to assess the preferred design approach for the system (selected as a result of the conceptual phase) against the system requirements. Once a sound system design is agreed to, the next step in the validation phase is to provide clear technical, contractual, economic, and organizational bases for full-scale development of the system (Glore, 1977:21). Thus, the major product of the validation phase is the Allocated Baseline (i.e., a set of preliminary development specifications, one for each CI/CPCI, which document the functional, performance, interface, design and testing requirements of the system as they are subdivided).

In developing the allocated baseline, the number and composition of CPCIs is a critical design issue. A system of many CPCIs has many formally defined interfaces. The separate reports, specifications, test plans/procedures and other monitoring activities required can support good Government visibility into, and control of, the development process. However, if a system is partitioned into too many CPCIs, the large number of document reviews, engineering change proposals(ECPs), and other management activities may fragment insight and cause excessive delays. Thus, the development of a multi-CI system may suffer more delay from Government monitoring activities than a system of fewer CPCIs. Somewhat different problems can arise if the number of CPCIs are few, but ill-defined. For example, a CPCI may contain processes that interact more strongly with other CPCIs than with one another. The inter-CPCI interfaces, although few, may also be very complex. As a result, the larger scope of the individual CPCI design reviews may fail to identify many inconsistencies among CPCIs. These oversights lead to many ECPs and to progressively more expensive repairs, depending on when each error is detected (Glore, 1977:26-27).

Although the author of this guidebook states that he knows of no well-defined procedure for specifying an optimum set of CPCIs, he does provide a list of guidelines that should help define a good set of CPCIs. In summary, they are as follows:

a. Assign processes that interact strongly to the same CPCI.

b. Assign processes that have little or no interaction to different CPCIs.

c. Allocate to different CPCIs processes that will execute in different computers.

d. Assign to different CPCIs processes whose development can feasibly be finished at significantly different times, if such phased development will expedite overall system development.

e. Allocate to different CPCIs software to be procured separately.

f. Include in each CPCI no more than a small, well-knit group of Government monitors can efficiently track, assuming reasonable working relationships between them and the types of personnel who will manage and develop the CPCI.

(Glore, 1977:27)

An Air Force Guide to Software Documentation Requirements (ESD-TR-76-159). This guidebook identifies and describes the most important Air Force technical and management documents relating to software acquisition. It is intended for managers and technical personnel who are responsible for determining the documentation requirements for software in a large system acquisition. In addition, general conclusions and specific recommendations for improving software documentation are presented. One of the conclusions specifically concerns the guidelines for CPCI selection. In summary, the author concludes that, "Determining the number of CPCIs is one management decision which has a heavy impact on software documentation requirements" (Schoeffel, 1976: 134).

Since the number of external CPCI interfaces increases as more CPCIs are defined and the documentation requirements is a function of the number of CPCIs, one approach to determining the most appropriate number of CPCIs is to minimize the CPCI interdependencies and interfaces. The author recommends that the number of inter-CPCI interactions for each alternative CPCI structure could serve as a figure of merit, along with some other suitability factors. For example, the other factors might include adequacy of visibility, degree of development control necessary, degree of criticality or risk involved, anticipated contractor capabilities (experience with equipment or specific functional areas), and expected frequencies of changes in specific areas. In addition to these criteria for evaluating the alternative CPCI structures, some of the previously cited guidelines for CPCI selection are reiterated in this guidebook. To summarize, they are as follows:

a. Define separate CPCIs for capabilities developed or delivered at different times.

b. Separate operational programs from utility programs and those from diagnostic programs.

c. Separate out particular programs with potential use in multiple systems.

d. Combine highly interrelated computer programs into a single CPCI.

(Schoeffel, 1976:134-135)

An Air Force Guide to Computer Program Configuration Management (ESD-TR-77-254). Of all the sources reviewed in this extensive literature search, this document provides the most comprehensive treatment of the CPCI selection process. It is written as a combined instructional and reference document which identifies, interrelates, and explains the requirements of current Air Force and DoD configuration management standards. Emphasis is placed on the acquisition of computer resources in major defense system programs. Specific background information, guidance, pitfalls, and examples are devoted to software elements of C^3 programs (Searle, 1977:1).

This guidebook defines the selection of configuration items as, "The process by which the complete set of equipment, computer programs, and facilities elements contemplated for a system as a whole are separated for purposes of managing their development or other procurement into individually-identified subsets." Hence, the configuration item is regarded as a level of management. Specifically, it is the level:

a. at which the procuring activity specifies, contracts for, and accepts individual parts of the system,

b. below which the developer is responsible for management of the development, or procurement, and assembly of item components, and

c. above which the procuring activity retains responsibility for interfaces, integration, and system performance (Searle, 1977:21).

Thus, the CPCI is the level at which a program office exercises formal management control over the responsible contractor in the areas of configuration management, procurement, program control, and technical progress.

The initial selection of CPCIs is usually accomplished at an early stage of the acquisition program. It represents a design decision and is basically a technical product of the system engineering process. In summary, the identification of a given assembly of computer program instructions and coded data as a CPCI results from the steps of:

a. functional analysis and definition of system performance requirements, and

b. system design, during which the defined functional and performance requirements are allocated among planned assemblies of system physical elements.

Therefore, the CPCI designation constitutes a commitment to develop a deliverable end item (e.g., in the form of a tape or deck of cards) which will perform its allocated functions when eventually assembled into the system (Searle, 1977:21-22).

Like several other studies and technical reports, this guidebook provides a "shopping list" of criteria to be used in selecting CPCIs. First of all, the author proposes that the intended source (contractor) is an essential starting point for decisions, since assemblies of computer program elements to be acquired from a single contractor are potentially a single CPCI, and assemblies to be acquired from separate sources must be separate CPCIs. Factors of cost, complexity of documentation, interface control, and other requirements dictate that it is generally desirable to avoid having any more CPCIs than necessary. Therefore, for a given single contractor, a productive approach is to start with the tentative assumption of a single CPCI, then "shredout" into separate CPCIs only when fully justified by an applicable criterion such as one of the following:

> a. Separate Computers. Computer programs to be designed for operation in different types or models of computers must be separate CPCIs.
>
> b. Separate Schedules. Computer programs scheduled for development, testing, or delivery at different times may be separate CPCIs.
>
> c. Different System Functions and Uses. In general, mission, support, and diagnostic (off line) computer programs should be separate CPCIs.
>
> d. Different Deployment Phase Control. Computer programs intended for different systems, and/or for different configuration control during the deployment phase, should be identified as separate CPCIs, even though they may be largely identical at the time of initial deployment and delivery.
>
> (Searle, 1977:23-24)

In summary, the author stated, "Although a single 'right' solution may not always present itself, reasonable care and attention to the considerations outlined above should yield sound results. On the other hand, success of the program can be almost precluded if those objectives and principles are disregarded" (Searle, 1977:24).

Software Acquisition Management Guidebook: Reviews and Audits (ESD-TR-78-117). This report combines existing guidance regarding reviews and audits currently contained in a number of official documents into a single document, and narrows the focus of existing guidance to those problems associated with software acquisition management. Detailed guidance is provided for the engineering design reviews which are conducted on the system during the validation phase to evaluate the CI/CPCI definitions and to determine if system design compatibility has been maintained (Neil, 1977:5-11).

During the early stages of the validation phase, a System Requirements Review is conducted to monitor the contractor's system engineering activities. For example, the refined system requirements and preliminary CI/CPCI selections are reviewed. These preliminary decisions normally result from initial trade-off studies conducted by the contractor. Although the initial CPCI breakout is done on a technical basis, the final decision must be tempered by management and acquisition issues. In fact, the author of this guidebook asserted:

> "Many acquisition problems are created if CPCI definitions are only viewed technically. A common misconception is that more CIs will provide more control and visibility. In fact, when many CPCIs are defined, costs are increased (increased data, management reviews, and control, and configuration management requirements), the program office accepts more responsibility (an increase in the number of development specifications, each with interfaces that must be approved by the Government when the specification is baselined), and the visibility remains essentially the same (the development specification requirements must still be defined at the performance level)" (Neil, 1977:19).

Conference Proceedings

In the past few years, papers addressing the acquisition and management of embedded computer resources have been presented at numerous conferences and symposiums. Unfortunately, most of them are primarily concerned with the technical aspects of software engineering and programming methodologies (e.g., requirements analysis, top-down versus bottom-up design, etc.). In other words, the process of allocating software requirements to CPCIs was rarely addressed although CPCI selection is a technical decision tempered with management considerations. This fact is somewhat disturbing considering the effect this decision has had on the overall success of many large weapon system acquisition programs. However, a few authors have discussed the need for better techniques for decomposing software requirements in general. For example, the presenter of one paper at the Second International Conference on Software Engineering concluded, "A technology needs to be developed that permits the structured decomposition of the system specification into a complete and consistent set of data processing subsystem requirements" (Belford, et al., 1976:72). This paper and the others relevant to this research are summarized in the following paragraphs.

Specifications: A Key to Effective Software Development. In supporting the above conclusion, the author of this paper states, "The initial steps in the subsystem engineering phase should include the decomposition of the (system) specification into discrete processes and the verification of these processes. Therefore, this decomposition process should trace the original system requirements specifications from one level of decomposition to another (vertical traceability) and also trace each of these requirements within any given level of decomposition (horizontal traceability)" (Belford, et al., 1976:72). This would hopefully provide the capability to identify inherent conflicts, discrepancies, and omissions and ensure the completeness of each requirement defined in the system specification. In summary though, the author admits, "The key issue remaining in the decomposition technology is performance allocation" (Belford, et al., 1976:79).

Managing the Development of Reliable Software. This paper, presented at the 1975 International Conference on Reliable Software, highlights a number of techniques and project activities which have proven to be especially valuable to the TRW Defense and Space Systems Group in their efforts to develop reliable software. In analyzing the "specifics-what have we done and what have we learned," the author related the following:

a. There is nothing, absolutely nothing, that can cause a more devastating outcome of a development activity than an incomplete and/or incorrect knowledge and corresponding specification of system and software requirements.

b. No matter how you slice it, the creation of a highly sophisticated, yet implementable design solution which will clearly satisfy demanding design, functional processing and performance requirements is far from an easy task.

c. We need to reduce the normal wait time before one could "hear the process talk." An incremental approach to detailed design, code, and test in conjunction with top-down concepts is possible to implement through dummy processes, loops, and builds. In other words, within that structure, it is possible to identify selected portions of the software (e.g., processing threads involving critical functions) which can be developed and tested as independent increments. With care, these increments (or loops) can be chosen to provide a complete increment of system capability and, thereby, substantially reduce the likelihood of last-minute surprises.

(Williams, 1975:2-4)

Requirements-Oriented Design: An Emerging Design Strategy. In this paper presented at the Spring 1977 COMPCOM, the author proposes that a user's needs may be broken into two categories: Requirements and Attributes. Basically, requirements are the constraints which the system must satisfy (i.e., what the system "must do"). Attributes, on the other hand, specify either options or evaluation criteria for qualitative comparisons of competing systems that meet the system requirements. For example, attributes may be used to evaluate competing architectures to obtain a feel for the "goodness" of the architecture in solving the customer's problem. Although some of them do not apply to software, several criteria are presented for use in determining the required attributes. They are: flexibility, expandability, bus complexity, executive complexity, availability, partitioning, modularity, reliability, maintainability, manufacturability, production cost, development cost, technical risk, logistics, programmability, support software cost, software adaptability and transferability, compatability, and service (Thurber, 1977:305-307).

Periodicals

Like the papers presented at various conferences and symposiums, articles in the numerous management and technical periodicals have seldom addressed the criteria for selecting CPCIs and the impacts of that decision on other aspects of the system program. However, incremental development has been the subject of a few articles. Similarly, some research has been performed on the criteria to be used in decomposing system requirements into modules. For example, D.L. Parnas proposed "information hiding" as a technique for decomposing systems into modules (Parnas, 1972:1053). Since the fifth objective of this research is to determine the feasibility and evaluate the potential impacts of defining CPCIs in terms of system versions or models (to be developed incrementally), the more relevant points in these articles are summarized below.

Perspectives on Software Engineering. As indicated by Marvin V. Zelkowitz in this article, each stage of the software development life cycle has its own set of problems and solutions. The most advanced techniques apply to the latter stages since the first stages are the least developed. For example, testing and debugging tools are apparent to every programmer since they are the oldest and most advanced. On the other hand, optimum design/development strategies are not readily available for each situation. However, the author does describe one approach which gives the user a running system early in the life cycle when changes are easier to make. This approach, called iterative enhancement (or incremental development), is another technique (like top-down development) for implementing hierarchically structured programs. In this technique, a subset of the problem is first designed and implemented. Then, the process is repeated to develop successively larger subsets until the final product is delivered. At each step of the process, not only extensions but also design modifications can be made. However fewer and fewer modifications are needed as the iterations converge to the full system. Thus, iterative enhancement can make rebuilding less chaotic since there is a running system early in the development cycle (Zelkowitz, 1978:207-212).

Iterative Enhancement: A Practical Technique for Software Development. The authors of this article, Victor R. Basili and Albert J. Turner, are generally credited with demonstrating this technique as a practical means of using a top-down, stepwise refinement approach to software development. In summary, their article, published in the IEEE Transactions on Software Engineering, in December 1975, states:

> The development and quantitative analysis of a production compiler for the language SIMPL-T was used to demonstrate that the application of iterative enhancement to software development is practical and efficient, encourages the generation of an easily modifiable product, and facilitates reliability (Basili and Turner, 1975:390).

However, the need for experimental, iterative design was recognized as early as 1973 when the CCIP-85 Study drew the following conclusion in Volume VII, entitled "Technology Trends: Integrated Design" of their report.

> One of the most recurrent themes of this volume is the notion using "experimental design" to determine explicitly, and on the most cost-effective basis, the optimum configuration and capabilities for a C&C system. Perhaps "experimental design" is an inaccurate description; "design through testing, modifications, and validation of increments of a system for maximum effectiveness" may be more apt (Boehm, 1972:44).

Textbooks, Essays, and Academic Papers

Many authors have contributed to our understanding of the problems associated with software development by presenting the theoretical aspects in textbooks, essays, and academic papers. All phases of the software development life cycle have been discussed in this academic arena. However, most of the emphasis has been on requirements analysis and system design. For example, automated techniques for analyzing system requirements, methods for decomposing the requirements into modules and subroutines, and design considerations such as cohesion and coupling have all received a lot of attention. As a result, only one publication relevant to the process of selecting CPCIs for the purpose of managing their development is referenced in this section.

Student Workbook for Course B: Software Acquisition and Management. This document was published by System Development Corporation and used as a handout in their Avionics Software Training Program. In the configuration management and identification section the following six criteria for CPCI selection are presented.

a. CPCIs are not part of an equipment CI.

b. Number of CPCIs and interfaces should be minimized.

c. Size is not a criterion.

d. Separate CPCIs should be defined for separate contractors.

e. Separate CPCIs should be defined for separate users/managers.

f. CPCIs are the basis for support documents.

(Searle, 1974:25)

Chapter Summary

As more and more of the processes which control our life-styles are automated, a higher level of trust is placed in the reliable functioning of this proliferating technology. During the past 10 years, the problems associated with automating these processes have been extensively studied by technical and management analysis teams. Thus, there is an abundance of information published concerning the acquisition and management of computer resources. This information emanated from various sources, and in most cases was directed at specific functional areas. Since the documents reviewed and summarized in this section were of the non-regulatory type, studies, technical reports, conference proceedings, textbooks, essays, lessons learned, periodicals, and guidebooks were included.

The process of allocating system software requirements to CPCIs was not the specific subject of any published information discovered in this research. However, several criteria and factors to be considered in the selection of CPCIs were presented in several publications. The more important ones are as follows:

a. Premature "freezing" of system structure into management structure causes problems (Ref. Para. a, p. 81).

b. Each operational program should be a CPCI.

c. Software tools should be separate CPCIs; however resident debug programs are part of the operation CPCI.

d. CPCIs should be small enough for one person to manage.

e. Decompose the system so that management control and flexibility can be maintained.

f. Partition the system so that implementation, debugging, and testing of modules or groups of modules (CPCIs) can proceed in parallel.

g. CPCIs should be identified based on whether they satisfy an end-use function, are key programs, have expected interface problems, and will provide added visibility to a particular computer program.

h. The CPCI selection decision should be made early in the program.

i. Care must be taken to include the AFLC system manager in the selection process.

j. The number of CPCIs should be minimized.

k. A CPCI must be operated and tested as a single computer system.

l. The end result of choosing a CPCI as a unit of management should be the ability to directly observe that a meaningful portion of the system works. Thus, a structure in which the "functional flow" across the system is divided into several CPCIs is artificial.

Several other management considerations were included in this chapter. For example, the risk, planned use, complexity, level of change control required, expected modification activity, and cost thresholds were all identified as factors bearing on the CPCI selection decision. In addition, the findings and recommendations of the Air Force Studies Board, Committee on Operational Software Management and Development, regarding the incremental development approach were summarized. Although this appears to be unrelated to CPCI selection criteria, it was included in this literature search since the fifth objective of this research was to determine the feasibility and evaluate the potential impacts of defining CPCIs in terms of system versions or models.

V. Results

This chapter presents and discusses the results of the analysis performed on the interview data collected in this research effort. The remainder of this chapter is devoted to presenting demographics of the sample population, characteristics of the development/acquisition program evaluated, CPCI selection criteria on that program, and the respondents' opinions on the feasibility, potential impacts, and implementation of horizontal allocation as a methodology for allocating software requirements to CPCIs.

Description of the Sample Population

The sample population, as described in Chapter II, included 45 respondents representing 10 Air Force organizations, 2 Federal Contract Research Centers, and 11 contractors of the U.S. Aerospace industry. Although the limited time available for the research did not permit interviews with a larger sample, the various backgrounds and areas of expertise reported by the respondents established a data base of facts and expert opinions that was assumed to be representative of the overall population. In addition, the sample was assumed to be sufficiently large to support the overall conclusions and recommendations made in Chapter VI. These assumptions appear to be reasonable in light of the descriptive statistics for the sample demographics.

Descriptive Statistics. Information on the distribution, variability, and central tendencies of each demographic is shown in Appendix G, Table VI. These descriptive statistics reflect a sample population that was highly educated, trained, and experienced. For example, 15 different software acquisition and management disciplines were represented in the sample. Likewise, the grade level for the respondents ranged from Captain to General for military personnel, GS-12 to GS-16 for civil service personnel, and programmer/analyst to vice-president for contractor personnel. Although only about half of the sample have had any formal training in either system engineering, software engineering, hardware engineering, or acquisition management, more than half of the respondents reported their educational level as completion of a masters degree.

The respondents were practically all (91.1%) currently working in the development/acquisition of computer software. Furthermore, the average respondent had been involved with these types of programs for more than 12 years. This experience was spread over more than 6 system programs. A large majority of the respondents (82.2%) indicated the acquisition and management of computer software embedded in C^3 systems was the primary emphasis of their experience. About one-fourth of the respondents reported having experience in the other major categories of software application. This diversity of system application experience was repeated in the type of software experience the sample population had. For example, more than half of those interviewed indicated they had experience in programming, systems analysis, and/or test and evaluation. Similarly, more than one-third of the sample population were experienced in software maintenance, system design, and/or acquisition management.

Analysis of Variable Interdependency. Some bivariate correlations among the demographic variables were statistically significant (Ref. Appendix J, Table IX). However, none of them were unexpected. For example, the number of system acquisition programs a respondent had worked on was correlated with the number of years of experience he had in the development/acquisition of software.

Program Characteristics

In order to evaluate the criteria presently being used by SPOs and contractors for allocating system requirements to CPCIs, the respondents were asked to identify a specific program, and then to answer a series of questions relating to those criteria. These questions included attributes of the program, identification of the selection criteria, advantages/disadvantages of the criteria, and problems with the CPCI structure of the program. Some of these questions asked for specific values, such as the number of CPCIs on the program. Others requested the respondent to provide a yes/no answer and then briefly expand on his answer. Finally, some questions asked the respondent to provide a subjective answer (e.g., what changes, if any, he would like to see in the way CPCIs are selected). The statistical analysis results for each program characteristic question that is quantifiable are presented in Appendices H (descriptive statistics) and J (statistically significant bivariate correlations). In addition to these statistical results, the qualitative responses are summarized in the following sections.

Program Identification? Although the scope of this research was limited to large software-intensive systems, 40 different system development/acquisition programs were identified and discussed by the respondents. A listing of these is attached as Appendix K.

Years Assigned to the Program (Q28)? The length of time the respondents worked on their identified program ranged from less than 1 year to over 10 years, with an average of 3½ years.

Respondent's Job on the Program? All activities involved in the system acquisition and software life cycles were represented in the job descriptions of those personnel interviewed.

When Were System Requirements Allocated to CPCIs? System requirements were allocated to CPCIs in most cases at the outset of each contract. Of the 40 programs reported on, CPCI selection was done 13 times during the conceptual phase and 16 times during the full-scale development phase.

The other answers were about evenly spread among validation phase, unknown, and not done yet. However, on one program the selection was performed as late as the critical design review(CDR). Furthermore, one respondent indicated that no allocation was accomplished until the contractor got in trouble during development.

Who Performed CPCI Selection? The initial allocation of requirements to CPCIs was performed by the contractor for more than half of the programs. In most of those cases, the contractor proposed the CPCI structure and the Government modified/approved it. However, Government and/or Federal Contract Research Center personnel did allocate the system level requirements to CPCIs on 13 of the 40 programs. In these cases, the CPCIs were defined either in the system level specification or statement of work, and then directed upon the contractor. Finally, only one respondent indicated that he was unaware of who selected the CPCIs on his program.

Program Development Cost (Q29)? The programs represented in this research ranged from less than $1 million to $500 million in total development cost. Although more than 25% of the respondents indicated that cost data was unknown or not available to them, a sufficiently large sample was available for application of statistical techniques. This analysis provided two interesting results. First, a large number of the programs were in the $100 million to $500 million category. Secondly, simple bivariate correlations between the program characteristic variables and the demographic variables indicate a positive correlation coefficient of 0.40 between experience in software engineering and total development cost. In other words, personnel assigned to the large programs have more software engineering experience than those assigned to the smaller programs.

Number of CPCIs (Q30)? A large amount of variance was reflected in the answers to this question. In other words, the number ranged from 1 to 200, with an average of 21 and a standard deviation of 27.069. In some cases, the number of CPCIs delivered to the user was considerably smaller than what was initially selected. One respondent explained this situation by stating, "Once the contractor got into trouble and development/integration couldn't proceed, the CPCI structure, which was forced on the contractor in a bizarre way, had to be redefined because functions were cutting across lots of CPCIs in an illogical fashion."

The simple bivariate correlations performed on the demographic and program characteristic variables revealed two interesting results. First, the number of CPCIs and a respondent's opinion as to the adequacy of the analysis performed on the alternative CPCI structures were negatively correlated with a correlation value of -0.37. This result suggests that an adequate analysis of the alternative CPCI structures might reduce the number of CPCIs on a software-intensive system. Secondly, there is a statistically significant positive correlation between the number of CPCIs and whether the respondent had any formal training in system engineering, software engineering, hardware engineering, and acquisition management. Therefore, more trained personnel were assigned to larger programs (in terms of the number of CPCIs) than to the smaller ones.

Number of Lines of Code (Q31)? Over 80% of the programs/projects represented in the interviews contained computer software that exceeded 100,000 lines of code (i.e., executable instructions). This result was expected since the initial plan was to only interview people with experience on software-intensive systems. In addition, the number of lines of code was significantly correlated with total development cost (0.75), number of CPCIs (0.62), and the respondent's opinion as to the adequacy of the analysis performed on the alternative CPCI structures (0.43).

Cost Ratio-Software to Hardware (Q32)? Only 47% of those interviewed answered this question; and several of those that did indicated their answer was strictly an estimate. Therefore, the validity of any conclusions based on the responses to this question is questionable.

Percent Contract Complete? The percentage of the software development/contract period that was complete ranged from 2% to 100%, with an average of 69%. In other words, all stages of the software development cycle, including 100% complete, were represented in the reported programs.

CPCIs Defined at Right Size (Q34)? A majority (62%) of the respondents answering this question believed that the CPCIs were defined at about the right size in terms of managing their development. This result was negatively correlated with the cost ratio of software to hardware at the 0.53 level. In other words, as the cost ratio of software to hardware increased, the respondents tended to believe that the CPCIs were not defined at about the right size. In addition, some of the problems that can occur when CPCIs are improperly sized were described by those responding negatively to this question. For example, one respondent indicated that too many CPCIs were selected; and as a result, excessive costs were incurred for documentation, configuration control, reviews, ECPs and test plans/procedures.

A more basic problem was highlighted by one Government expert who stated,

> "The CPCI structure was meaningless. It did not represent a meaningful decomposition of the system. CPCIs violated the intent of AFR 65-3 (i.e., they did not meet the 'end-use function' criterion). As a result, a CPCI that worked meant nothing in terms of system performance; hence, they were useless in obtaining management visibility."

In addition, several respondents did not consider size to be a primary concern in CPCI selection. One contractor expert made this point clear by stating, "Size is only used by the technical man who doesn't concern himself with the contract."

Familiar with CPCI Selection Criteria (Q35)? Over 70%

of those responding to this question indicated that they were familiar with the criteria used for allocating system requirements to CPCIs on their present (past) program. In addition, they listed many of the criteria. Those most often cited by the respondents were:

a. "Functional modularity (i.e., grouping like functions) is a necessity."

b. "Software developed by separate contractors/vendors shall be separate CPCIs."

c. "Support software shall be a separate CPCI."

d. "Each CPCI shall operate on a single computer."

e. "Minimize the number of CPCIs and interfaces."

f. "Keep the complex interfaces within a CPCI."

g. "On-line and off-line diagnostics shall be separate CPCIs."

h. "Similar requirements and functions that operate interactively shall be in the same CPCI."

i. "Off-the-shelf and developmental software must be separate CPCIs."

j. "Separate locations and functions require separate CPCIs."

In addition, some other considerations were mentioned less frequently by the respondents. They are summarized below:

a. "It should be possible to effectively test a CPCI as a whole."

b. "Selection should be based upon the CPCI end-item support requirement (i.e., method of distribution)."

c. "Allocate on the basis of system performance parameters, such as MADT(maximum allowable down-time) and Pt(processing time)."

d. "Politics among the contractors, procuring agency, using organization, and maintenance organization must be considered."

e. "Previous experience with a similar type of system is desirable."

f. "User and maintainer advocacy must be considered."

g. "Documentation requirements should be minimized."

h. "Separate users/deployment phase control should signal separate CPCIs."

How Were the Criteria Selected? Although almost half of the respondents were unfamiliar with how the CPCI selection criteria were chosen for their program, a wide variety of answers were received for this question. They ranged from "the contractor did it" to "the Government imposed them," with "previous experience" being the most often cited response. In addition, the following considerations were identified:

a. "Comparisons with capabilities of the existing systems"

b. "Efficiency and ease of documentation"

c. "Contractual responsibilities"

d. "Delivery schedules"

e. "Transfer and turnover"

f. "Common data base usage"

g. "Number of design reviews and audits"

h. "Requirements traceability for testing"

i. "AFLC responsibility for the operating system and off-line diagnostics"

j. "Vendor documentation copyrights and updates"

Despite the listing of some criteria in the DoD and AF policies and procedures on software acquisition, only two responses mentioned these guidelines. On the other hand, several non-technical responses were received. They included, "common sense," "uneducated guess," "kludge," "by default due to inexperience and lack of software acquisition understanding," "'blue sky' and 'back of envelope' calculations," and "judgement of the people preparing (or managing) each system segment specification."

Advantages and Disadvantages of These Criteria? Several advantages and disadvantages of these CPCI selection criteria were listed by the respondents. They were categorized by the method of their selection. In general, there was agreement that those system requirements allocated by the contractors resulted in a total collection of system software that can be managed in accordance with established Air Force procedures. Although the number of respondents in this category was small and primarily consisted of contractors, several specific advantages were described. They included the following:

a. "The Government can hold the contractor responsible for a totally integrated design package."

b. "A complete package is available at acceptance."

c. "Cost to the Government is minimized."

d. "The Government does not accept responsibility prematurely."

e. "Programmer assignment is easier."

f. "Unit testing is easier."

g. "Documenting the system is easier."

h. "The overall system is easier to define."

On the other hand, one respondent (a contractor) stated, "Proper application (of the selected criteria) sometimes requires more knowledge and experience in system acquisition concepts and implications than is always brought to bear." This problem was supported by another respondent who asserted that, "The criteria are not sufficiently documented in today's MIL-SPECs and are not understood by the personnel doing the selection."

For the "Government-imposed" criteria, the disadvantages appear to greatly outweigh the advantages. In other words, only three minor advantages were discernable by the respondents. They were:

a. "Traceability of requirements to design was made easier,"

b. "CPCIs were more manageable during development and test," and

c. "line item delivery was possible."

Conversely, several respondents described drastic results which occurred when CPCI structures were imposed upon contractors. For example, on one program, "The system level specification submitted to prospective bidders specified the computer program components(CPCs), as well as the CPCIs. As a result, several engineering change proposals(ECPs) were generated and submitted by the eventual bid winner." In another case, "Completion of the FQT(functional qualification test) did not allow observation of a working portion of the system. Also, the structure, when combined with the AFR 800-14 definition of allocated baseline, caused the SPO (systems program office) to focus on design detail and forget system performance."

Even though experience was the most often used methodology for choosing CPCI selection criteria, the respondents did not agree on its utility. For example, one respondent stated, "Experience may lead to selection of CPCI content that is forced into an unnatural mold because 'it was done this way before' or because the people in charge have no knowledge of better techniques." Another respondent indicated that CPCI selection criteria chosen on the basis of experience are hard to clarify, and often overlap. This results in "independent systems, duplicated functions, and difficult integration problems." On the other hand, criteria chosen in this manner do "minimize specification maintenance problems (i.e., ECPs), testing/requirements traceability problems, and turnover problems."

The two respondents who reported on criteria selected from the DoD and AF policies and procedures (i.e., MIL-STD-483 and an ESD Guidebook) stated that the criteria were "understandable and worked for most applications"; but "maybe all configuration contingencies were not considered."

Criteria Different from Those Previously Used (Q36)? Almost 40% of those interviewed failed to respond to this question; and those who did were evenly split among positive and negative responses. In most cases that were different, the previously used criterion was to allocate requirements on the basis of functional modularity. For example, CPCIs were defined for utility, control, application, simulation, and data reduction functions. However, one respondent, who had 18 years of experience in software development, stated,

> "In two (previous) cases, SPO advisors insisted that 'size' and 'visibility' were criteria to be applied with priority over all others. Generally, those particular advisors happened to be software-technical people with limited knowledge/experience in (1) the Air Force and (2) defense system program. They did not really understand the configuration management processes of identification, control, and status keeping, nor the ways in which those processes depend fundamentally on sound principles of CI selection. Both of those programs proved to be recognized failures. While there were other factors, in both cases, that approach to CPCI selection was sufficient in itself to ensure that serious troubles would be encountered, downstream."

Advantages and Disadvantages of Those Previously Used Criteria? Although CPCIs defined on the basis of functional modularity generally provide the manager with a "high level of visibility during the early development stages," this criterion has some significant disadvantages. They include:

a. "Documentation is a nightmare (e.g., many ECPs in the inter-CPCI interface area),

b. traceability of functional requirements to design is poor, and

c. suboptimization of design at the CPCI level is encouraged (e.g., each CPCI may satisfy its allocated requirements but software integration is still an extended process)."

Any Other Criteria That Should Be Used (Q37)? Over 60% of those responding to this question identified other criteria or considerations that they believe should be used in decomposing the system requirements. They are summarized as follows:

a. "Organizational interfaces associated with the development team must be considered. Ideally, the development team should be organized about the functional requirements of the job; but seldom is this ideal achieved."

b. "Interfaces between CPCIs should be as simple and stable as possible. If the interface is really a hairy one; then, perhaps, the CPCIs on each side of the interface should be done by the same contractor or the CPCIs should be made a single CPCI thus eliminating the interface."

c. "In a distributed processing environment, where microprocessors implement individual functions, there is a strong case for identifying CPCIs in accordance with the specific processor (computer) in which it will operate. This would result in a 'cleaner,' more manageable effort, especially in documentation."

d. "The future organizational structure of the user and maintainer should be considered. In other words, separate CPCIs are required if the future maintenance organization for the mission software is not the same as that for the support software."

e. "More careful consideration of the contractual and legal implications of a 'CPCI' are required. Respondents from both the Government and industry voiced this concern in their response to this question. For example, one Government expert described the following scenario. "We could impose the full vigor of AFR 800-14 and be in a position of having to accept individual CPCIs which satisfy the individual specifications and tests, but which when integrated among themselves and with the rest of the system could require major rework to achieve system performance. Our tendency has been to hold contractors to system performance but still require CPCIs. This seems like a contradiction and probably is. When a project goes well, no one cares about the contradiction; however, when a project goes to 'disasterville,' there seems to be the legal argument that the AF owns the CIs, thus it is not the fault of the contractor that he cannot integrate them to achieve system performance."

f. "Software requirements must evolve along system functional lines to minimize the communications gap between management, system engineering, and software engineering. The iterative evolution is important when the system is different from the teams previous experience."

g. "More emphasis on software maintenance is required. Good on-line and off-line diagnostics must be specified (as CPCIs). In summary, a contractor who designs (structures) a system for easy software maintenance will have a much easier time in the integration phase."

h. "More involvement of programmers in system design is desirable."

i. "A system must be decomposed into 'subsystems'-not in the sense that a DBMS is a subsystem, but in the sense that if the program were cancelled right now, would the CPCIs that have passed FQT give the user the ability to do some portion of his mission. CPCIs structured around (functional) system tasks (observation collection, math subroutines, etc.) would not."

In summary, one FCRC expert, who had 25 years of experience in the development/acquisition of computer software, stated, "This lessons-learned set of information is needed and may only come as a composite of responses to your survey."

Criteria for CPCIs Different From CIs (Q38)? Slightly over half of those responding to this question indicated that the criteria used for selecting CPCIs are different from those used for allocating system requirements into CIs. One respondent summarized the difference by stating, "CPCIs and CIs are both a level of delivery on a contract, but CIs tend to be delineated by physical criteria more so than functional or logical considerations." In other words, "Software hasn't concerned itself with F^3(form, fit, and function), interface standardization, etc. like hardware has. As a result, there are no known MIL-STDs for software system design or software modules as there are for hardware." In summary, "The basic objectives and essential considerations are the same for CIs and CPCIs. However, there is more flexibility in the case of CPCIs, due to the absence of requirements associated with production/manufacturing, spare parts control, etc."

Experienced Any Problems With the CPCI Structure (Q39)? Over 72% of those responding to this question indicated that they had experienced problems with the CPCI structure on the program they evaluated. In other words, evidence exists to support the hypothesis that problems are experienced with the CPCI structure on more than half of the system development/acquisition programs that are software-intensive (Ref. Appendix L for t test results). Another interesting statistical inference is that problems with the CPCI structure become more prevalent as a program gets larger. This was indicated by the simple bivariate correlation of -0.55 between this question and program size in terms of number of lines of code.

As reported by several respondents, the impact of these problems can be devastating to a system development/acquisition program. In many cases, "documentation becomes confusing, requirements are 'lost,' schedules are slipped, users get frustrated, and CPCIs are delivered with an enormous number of software discrepancies." The solutions to these problems have been, in some cases, as drastic as the problem itself. For example, one "program was cancelled" and in another one "the schedule slipped 18 months, costs ballooned, and testing requirements were relaxed." However, in most cases, "ECPs are generated to modify the initial CPCI list and thousands of difficult software fixes are performed."

Alternative CPCI Structures Evaluated (Q40)? Over half of those responding to this question did not believe that an adequate analysis of the alternative CPCI structures was performed on their program. The comments supporting this result ranged from, "There were no real alternatives considered-we should have but we didn't," to "In the framework of system programs managed in accordance with established Air Force policy, there are no alternatives." However, the majority of those responding negatively identified preliminary system design as the area needing additional emphasis. For example, "More technical, management, and operational planning are required before CPCI boundaries are fixed." One

Government expert clearly described this need by stating, "The current thrust of analysis (per AFR 800-14) is hardware oriented (e.g., through-put, core, timing, etc.), rather than oriented towards a logical decomposition of the system into manageable units, which if managed properly, will result in a working end-use system function vice an internal non-stand alone system task."

The simple bivariate correlations between the responses to this question and the other program characteristics questions produced some interesting inferences. First of all, the respondents' opinion on the adequacy of the analysis of alternative CPCI structures was negatively correlated with program size, both in terms of number of CPCIs (-0.37) and number of lines of code (-0.43). In other words, those personnel evaluating large software-intensive programs tended to believe that an adequate analysis of the alternative CPCI structures was not performed on their program. However, if a respondent believed that an adequate analysis of the alternatives was performed on his program, he tended to agree that the CPCIs were all defined at about the right size in terms of managing their development. This was indicated by a positive correlation of 0.48 between these two questions.

In another case, the respondents' opinions on the similarities of CI and CPCI selection criteria were negatively correlated with their opinions on the adequacy of the analysis of alternative CPCI structures (-0.57). However, the most significant inference was that an inadequate analysis of the alternative CPCI structures leads to problems with the chosen CPCI structure. This was derived from a negative correlation of -0.52 between the respondents' opinions on the adequacy of the analysis of alternative CPCI structures and whether they had experienced any problems with the CPCI structure on their programs.

Suggested Changes in the Way CPCIs Are Selected. Many changes in the way CPCIs are selected were suggested by those interviewed. Most of the suggestions involved procedural concerns and/or the level of emphasis placed on this process. In addition, several respondents discussed the problems associated with selecting CPCIs on a software-intensive system without making any specific recommendations. One FCRC expert stated,

> "The CPCI structure is established too early in the acquisition cycle. More pre-RFP analysis is needed. More time is needed for government analysis before releasing contracts. The SDR, PDR, and CDR schedules must be slowed down to prevent the current practice of starting to code before PDR is over (SDR 90 days after contract award is silly). We rush too much early in the game, don't test enough; and pay dearly for it later. We always do-we never learn!"

This opinion was supported by several government and contractor experts who indicated the need for more up-front systems planning and an evolutionary approach to the development of large software projects.

The other important but less frequently mentioned changes were as follows:

a. "Isolation of vender software into separate CPCIs should be mandatory."

b. "Agencies responsible for the software maintenance after turnover should be identified and considered."

c. "More emphasis on the system product after integration and less emphasis on individual CPCI development visibility is needed."

d. "The problems of common/adapted software documentation needs more emphasis."

e. "The using and supporting activity must be involved beginning at the time the system is determined to become digital/computerized, and they should have responsibility, with the contractor, for CPCI selection."

f. "There is an urgent and widespread need for better training of both SPO and contractor personnel in the meaning, purposes, functions, and management implications of the CPCI concept."

g. "CPCIs should be selected on a functional string basis so that CPCI interrelationships are minimized."

h. "The discrepancies in AFR 800-14 and MIL-STD-483 regarding definition of the allocated baseline need to be resolved. The allocated baseline is supposed to be a 'design to' baseline, yet the content required by these documents is detailed design information. AFR 800-14 says 'math models' should go in Section 6 of the Part I specification (which is not contractually binding), then says 'algorithms' should be in Section 3. What is the difference; they should all be in Section 6, if in the Part I at all."

<u>Suggestions for Improving the Software Requirements Decomposition Process (Q41)</u>. This question provoked a wide variety of responses, which were categorized into five areas: visibility, structured approach, lack of understanding, expertise required, and flexibility. As for visibility,

one respondent stated,

> "CPCI selection must be made bearing in mind that the SPO is seeking visibility into contractural processes, and the CPCI structure is one tool to provide visibility. It must be recognized that meaningful visibility is not necessarily gained by taking 'vertical snapshots' of the system. Each facet of requirements decomposition must be accompanied by the questions-will this particular entity give me visibility, will subdividing more give me more visibility, and if each unit (CPCI) passes its FQT, will I have a complete (in the 'horizontal' sense) portion of the system up and running?"

Several respondents discussed the need for a more structured approach to the decomposition process. Both requirements definition methodologies and design techniques were included. For example, system simulation, top-down modular design, step-wise refinement, Structured Analysis and Design Technique(SADT), Computer Aided Design and System Analysis Tool(CADSAT), and computerized text editing were all suggested as tools that would help to better clarify the functions and their interrelationships.

Although the concept of partitioning the requirements of a system into subsystems, and a subsystem into components has been in use for many years, several respondents to this question believe that a general lack of understanding exists both in the Government and in the computer industry. A well-known contractor expert, who had 25 years of experience in the development/acquisition of software, stated,

> "I believe we have taken a major step backwards, since 1969 (when the decision was made to establish a standard approach for all three services). The goal was an admirable one but the results have not been that great. Having to re-write the MIL-SPECs and get tri-service agreement on them has resulted in all the detail being removed. Personnel entering the field since 1969 have great difficulty in understanding the process if they have to rely on today's MIL-SPECs and regulations. The MIL-SPECs assume that the reading audience has an in depth level of experience and knowledge which is non-existant in both government and industry."

Several approaches to minimizing the impact of this situation were suggested by the respondents. They included the following suggestions:

a. "Develop a good example of the requirements decomposition process and its documentation and make it available to all concerned parties."

b. "Fund the development of an acquisition guidebook, and incorporate the guidebook information in one of the AFSC 800-series pamphlets."

c. "Publish the results of this study of the process and make it available to the software industry, as well as those in the Government (procurers, developers, and users).

d. "Select a contractor who has people with a proven track record. Quit bringing in line pilots and navigators to run software programs. Let the people who have done it and learned from their mistakes, when they were young, try it. Use top-down approach. Get something working, then add capability. Hire companies that have a profit motive to perform analyses and studies, rather than an FCRC organization."

e. "Use personnel who have expertise in four areas:

(1) The prime functions allocated to the system whose CPCI requirements are being decomposed.

(2) Associated functions of the hardware CIs of the system.

(3) Immediately relevant functions of interfaced systems.

(4) The operational capabilities which will be provided by the system and how they will be employed."

f. "Provide more and better training of software engineers involved in system programs. Ensure more personnel are trained in the support management disciplines (notably, configuration management) with respect to principles and procedures that apply specifically to software."

Finally, flexibility seems to be a necessity, especially in developing any real-time system such as a C^3 system. In other words, "Requirements are going to change and enough flexibility should be planned into the project to allow for justified changes." This includes schedule and budget, as well as the system architecture.

Evaluation of Horizontal Allocation

Section III of the interview questionnaire asked the expert's opinion on the use of horizontal allocation as a methodology for allocating software requirements to CPCIs. To ensure a common understanding among those interviewed, a definition of the concept and a simple example were included with each questionnaire. Then, five questions addressed the feasibility, potential impact on 12 parameters, advantages/disadvantages, and implementation of the concept. The statistical results (tabulated in Appendices I and J) and qualitative responses for each question are summarized in the following paragraphs.

Feasibility (Q42)? Over 82% of those responding to this question indicated that horizontal allocation is a feasible methodology for allocating software requirements to CPCIs on a large software development project. This is sufficient evidence to statistically support the hypothesis that more than half of the software development/acquisition experts in Government and industry believe that horizontal allocation is feasible as defined (Ref. Appendix L for t test results).

The analysis of variable interdependency between this question and those nominal-scale variables identified in Appendix F (using the Chi-square test) resulted in no statistically significant correlations at or below the 0.05 level (Ref. Table XVII of Appendix J).

On the other hand, an analysis of the bivariate correlations between this question and those interval-scaled program characteristic questions produced some interesting interdependencies (Ref. Table XV of Appendix J). For example, the size of the program (in terms of number of CPCIs and lines of code) was positively correlated with the respondent's opinion on the feasibility of horizontal allocation. Likewise, those respondents who experienced problems with the CPCI structure on their program were more likely to agree that the methodology is feasible.

Similarly, the Pearson Product-Moment Correlations between the answers to this feasibility question and the perceived effectiveness questions were all positive and statistically significant, except for one (Ref. Table XVI of Appendix J). That is, the respondents' opinions on the feasibility of horizontal allocation and how effective it would be in reducing the software integration task were not significantly correlated.

Perceived Effectiveness (Q43-Q54). This question requested the respondent to evaluate the horizontal allocation methodology on the basis of its impact on 12 parameters. Each response was scored on a simple 1 to 4 scale, where 1 equals "not effective," 2 equals "somewhat effective," 3 equals "moderately effective," and 4 equals "very effective." The statistical results for each of these parameters are tabulated in Appendices I, J, and L. In addition, the results are briefly discussed in the following paragraphs.

An analysis of the responses to these questions provides two interesting results. First, a large majority of those responding to these questions indicated that the horizontal allocation of software requirements to CPCIs would favorably impact all but one of the 12 parameters (Ref. Table VIII of Appendix I). In fact, there was sufficient statistical evidence to support the hypothesis that more than half of the Government and contractor experts believe that horizontal allocation would be at least somewhat effective in favorably impacting all the parameters, except for complexity of the decomposition process (Ref. Table XIX of Appendix L for t test results). In addition, over 60%

of the respondents indicated that this approach to CPCI selection would be moderately to very effective in:

a. providing the program manager and user with more objective visibility into the system development,

b. improving the software quality (e.g., fewer latent errors),

c. reducing the software integration task, and

d. highlighting problems earlier.

These results were supported with written comments from some of the respondents. For example, one government expert stated, "The user loses phony visibility and gains visibility into a working system through 'seeing, touching, and feeling' the working increments." Another respondent wrote, "This approach to CPCI selection is basically intended to provide the user with some initial operational capability instead of having to wait on some longer schedule for the total operational capability." Software maintenance will also be easier and less costly according to one government expert who said, "The system will have been subjected to many more hours of realistic testing prior to turnover, thus less errors will remain. Similarly, if the contractor is required to write a product specification on each increment, and the specification is baselined immediately after the incremental FQT, the system will be documented very well, which is more than can be said for many of today's operational systems."

In addition, several respondents provided positive comments on the effectiveness of horizontal allocation in highlighting problems earlier. One government expert's response was, "Definitely. The real problems of a software development (i.e., the software doesn't work as a system) are not discovered in today's approach until after all CPCIs are coded and FQT'd. In this approach, system problems will be identified very early, and can be diagnosed on the spot."

On the other hand, some comments indicated that horizontal allocation could be detrimental to a software development project. One respondent indicated that, "This methodology would allow development to be accomplished in a structured step-by-step manner, rather than in one giant leap as we do it today. Therefore, the total schedule would conceptually be longer but most likely shorter since this development approach would be less risky." More specific comments were received on the impacts this approach would have on software integration. For example, one respondent stated, "This approach actually increases the amount of integration work by forcing system integration from the second FQT on."

Finally, the bivariate correlations among the 12 perceived effectiveness variables, and between the perceived effectiveness variables and the demographics/program characteristics produced several significant interdependencies. For example, the respondents' opinions on the effectiveness of horizontal allocation in making debugging and testing more efficient, making software maintenance easier and less costly, increasing the morale of those working on the program, reducing the complexity of the decomposition process, and improving training effectiveness were all positively correlated with the number of CPCIs on those programs evaluated.

In another case, the perceived effectiveness of horizontal allocation in reducing the software integration task was significantly correlated with the answers to the question, Have you experienced any problems with the CPCI structure on your present (past) programs? In other words, respondents who have experienced problems with past CPCI structures believed that the software integration task would be reduced on programs where CPCIs are selected on the basis of horizontal allocation. All of these statistically significant correlations are presented in Tables XII, XIII, and XIV of Appendix J.

Adversely Impact Contractor (Q55)? More than half of those responding to this question indicated that this technique would not adversely impact the contractor's/developer's management procedures. In addition, practically all of the respondents provided a brief explanation of the rationale for their answer. Several significant comments (both pro and con) were included in the explanations. Those indicating that horizontal allocation would adversely impact the contractor/developer included:

a. "A 'build' is a partial system. Therefore, a contract structure is needed to support iterative development and/or partial deliveries, such as a DD-250/payment for each 'build.'"

b. "Division of labor would be difficult."

c. "The contractors' management procedures are ultimately based on cost traceability with breakpoints (PDR/CDR/FQT/System test) imposed by the Government. The pieces implied by this approach are too small for the cost expended in tracking the development."

d. "The contractor would find this restriction on his organizational freedom a handicap in properly staffing the job."

e. "The developer must push himself to proceed to the next block, for it would be easy to continue testing forever."

f. "There would be a tendency to defer the really tough problems to future versions."

On the other hand, more than half of the comments indicated that this technique would benefit, not hinder, the contractor's/developer's procedures. Some of the more specific ones are listed below.

a. "This technique should improve the management procedures by providing more rapid feedback on progress as each build is completed."

b. "Assessment of development status would be alot more obvious."

c. "The impact will be towards less procedure oriented acquisition and more towards meaningful acquisition milestones and realistic system integration and test efforts."

d. "Incremental development, which results from this approach to CPCI selection, has been used on one USAF Strategic Air Command(SAC) program and it worked. It was implemented through a basic contract and several options which were to be exercised periodically. This approach gave the Government more control and made the contractor more responsive."

e. "This technique would force more advance planning and preliminary design since the CPCI structure becomes more important contractually. For example,

> the request for proposal(RFP) would probably require that CPCIs be selected in accordance with some guideline, standard or technical report; and then the CPCI structure would be a weighted evaluation factor (both technical and management) at the Source Selection Evaluation Board(SSEB)."

Other Advantages/Disadvantages (Q56)? Over 80% of those responding to this question identified some other advantages and/or disadvantages of selecting CPCIs on the basis of horizontal allocation. The major advantages were stated as follows:

a. "This approach should smooth out the use of contractor personnel and provide much better continuity of the design/implementation/checkout process. This leads to less risk for the contractor and the Air Force."

b. "The clear advantage is flexibility in the development, debug and implementation phases. Thus, it will simplify 'crash' diagnosis, system restart, and recovery." In addition, "Operational use of the first build can feed required operational modifications to subsequent builds."

c. "This approach, which naturally leads to incremental development, provides more meaningful milestones, real visibility into contractual status, earlier user involvement, less meaningless paperwork, and better informed management. Therefore, the visibility gained will not be at the price of causing the acquisition to become bogged down in bureaucratic battles over typos in Part I specifications. In addition, this technique allows airtight configuration control of the product as it evolves, rather than paper intensive, bogus control of the untested design via extant configuration management procedures."

d. "It forces the designers (often invisible or unknown to the implementer) to think functionally and from the start to finish."

e. "Usually programs are underestimated in terms of cost and time. This method should provide a better idea of the ultimate cost of the system and how long the development will take. As a result, the

Government may be more likely to cancel a program early, rather than trying to bail it out."

f. "This technique would help functional test planning and test documentation." In addition, "The two dimensions of test (capability-does it do the job?, and performance-how well does it do the job?) can both be accomplished."

g. "Although there would seem that no progress is being made initially, it will alleviate the traditional sell-off crisis."

The other disadvantages described by the respondents were primarily concerned with obtaining acceptance among the Government and industry, and how to contractually implement horizontal allocation. For example, one respondent indicated that, "There will be a huge psychological shift required to implement this approach. Most people do not understand." This point was illustrated by another respondent who stated, "The disadvantage appears to be the design of yet another regulation that will be out of date when approved. Circulation of data derived from bad experiences is great and should be encouraged. However, a regulation in this area, particularly with regard to a philosophical rather than purely technical viewpoint, does not appeal to me."

In the area of contractual implementation, the concerns expressed included the following:

a. "It may be difficult, if not impossible to write a fixed price contract for the whole development."

b. "Once you accept the first model, you would have to GFE(government-furnished equipment) it back to the contractor to build the next model."

c. "This approach may be overkill for a job that is well understood and similar to one previously performed by the contractor's team."

d. "The sole source problems currently experienced by many programs may become worse."

Should Horizontal Allocation Be Implemented (Q57)? Over 75% of those responding to this question indicated that this technique should be implemented on future software development/acquisition programs. In other words, a large majority of the respondents believe that CPCIs should be defined in terms of system versions or models, each of which contains end-use system functional capabilities. In fact, there was sufficient statistical evidence to support the hypothesis that more than half of the Government and contractor experts believe that this approach should be implemented (Ref. Table XIX of Appendix L for t test results).

The analysis of variable interdependencies produced some interesting results. First, no statistically significant interdependencies (at or below the .05 level) were found between this question and the nominal scale variables (Ref. Table XVII of Appendix J for Chi-square test results). On the other hand, the respondent's recommendation on the implementation of this technique was significantly correlated with the number of CPCIs (0.46) and the number of lines of code (0.48) on the program evaluated by the respondents. In other words, respondents who evaluated large programs believed that this approach should be implemented on future software development/acquisition programs.

Another interesting statistical result was that the respondent's opinion was significantly correlated with all the perceived effectiveness variables at or above the 0.35 level (Ref. Table XVI of Appendix J). The strongest relationships were between the implementation question and the respondent's opinion on the effectiveness of horizontal allocation in reducing system development costs (0.69), reducing system development time (0.58), making software maintenance easier and less costly (0.60), and increasing the morale of those working on the development program (0.58).

The respondents also provided several comments explaining their recommendation on the utility of horizontal allocation. Some of the comments were philosophical and/or general in nature, while others directly addressed ways of implementing this approach. For example, one respondent stated,

> "Management of the development phase in units, called builds is great, and even delivery in builds is great, but after delivery, you need to control each component (program) separately (e.g., operating system, application program, data base, and simulator). In other words, by defining CPCIs traditionally (Ref. Figure 7), nothing is preventing the development/delivery/installation of the system in builds (Ref. Figure 8)."

However, drastic problems often result from this approach. In other words, contracting for/managing a software development project on a basis different from that which the software is coded and delivered can be devastating to the total cost and schedule of the system. In fact, on one recent C^3 system acquisition, the identity of CPCIs defined in the

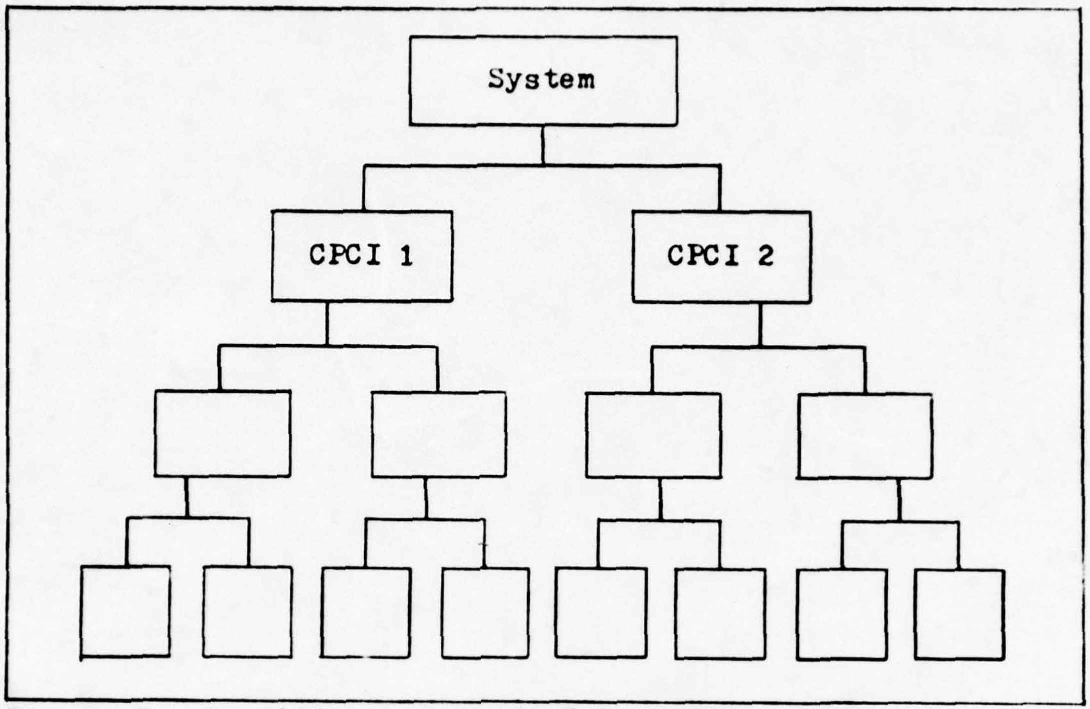


Figure 7. Traditional Method of Designing, Developing, and Delivering Software

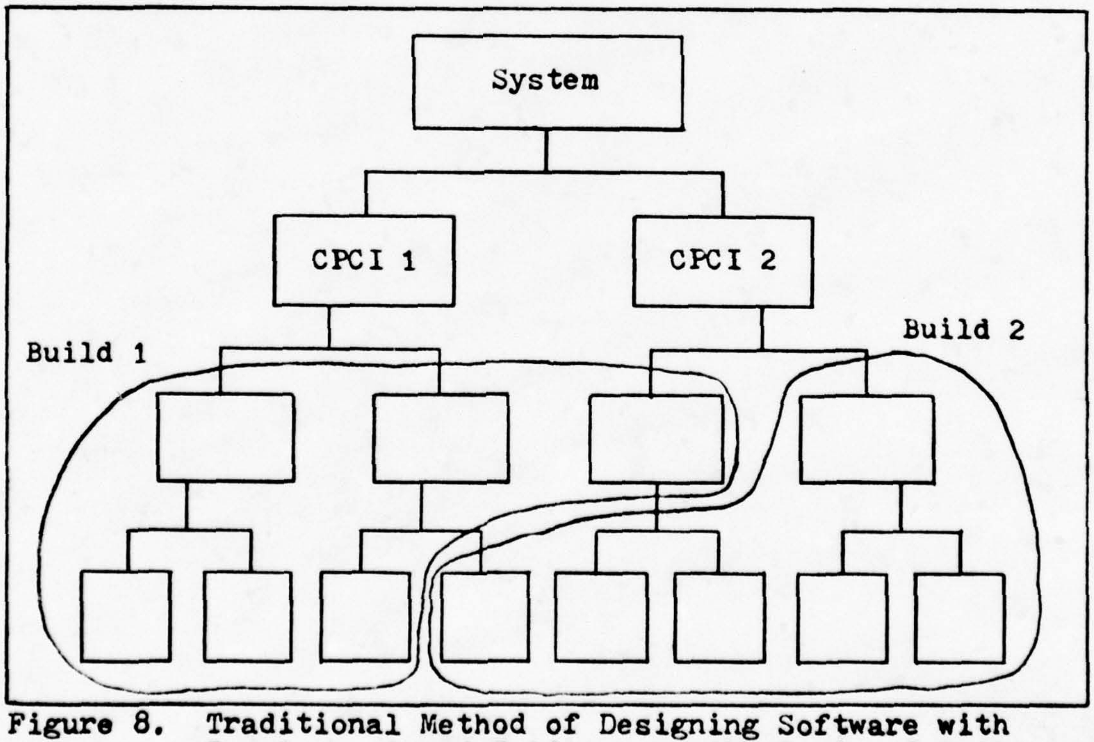


Figure 8. Traditional Method of Designing Software with Development and Delivery on the Basis of Builds

traditional method was lost when the contractor went to the "builds" approach in order to get a meaningful portion of the system working. This created difficult management and technical problems for the contractor, procurer, user, and maintainer. This same result was reported by several respondents.

On the other hand, one contractor expert indicated that his company has successfully used a top-down structured development approach where "builds" were defined vertically across all functions. Furthermore, he stated, "This is the best approach yet invented to guarantee a minimum of integration problems at each level. However, what is needed is more emphasis on the management of this approach." That is, "Horizontal allocation should be encouraged by the standards and regulations as a method of defining, developing, and delivering CPCIs (Ref. Figure 9), but must be left to the contractor to propose if he thinks it is appropriate to his project."

Another expert stated, "The value of this technique is that it provides for the necessary learning required by each new development team (i.e., one not having done a very similar project)." Thus, "Horizontal allocation should be selectively implemented on a medium to small sized program to determine if fine-tuning is required."

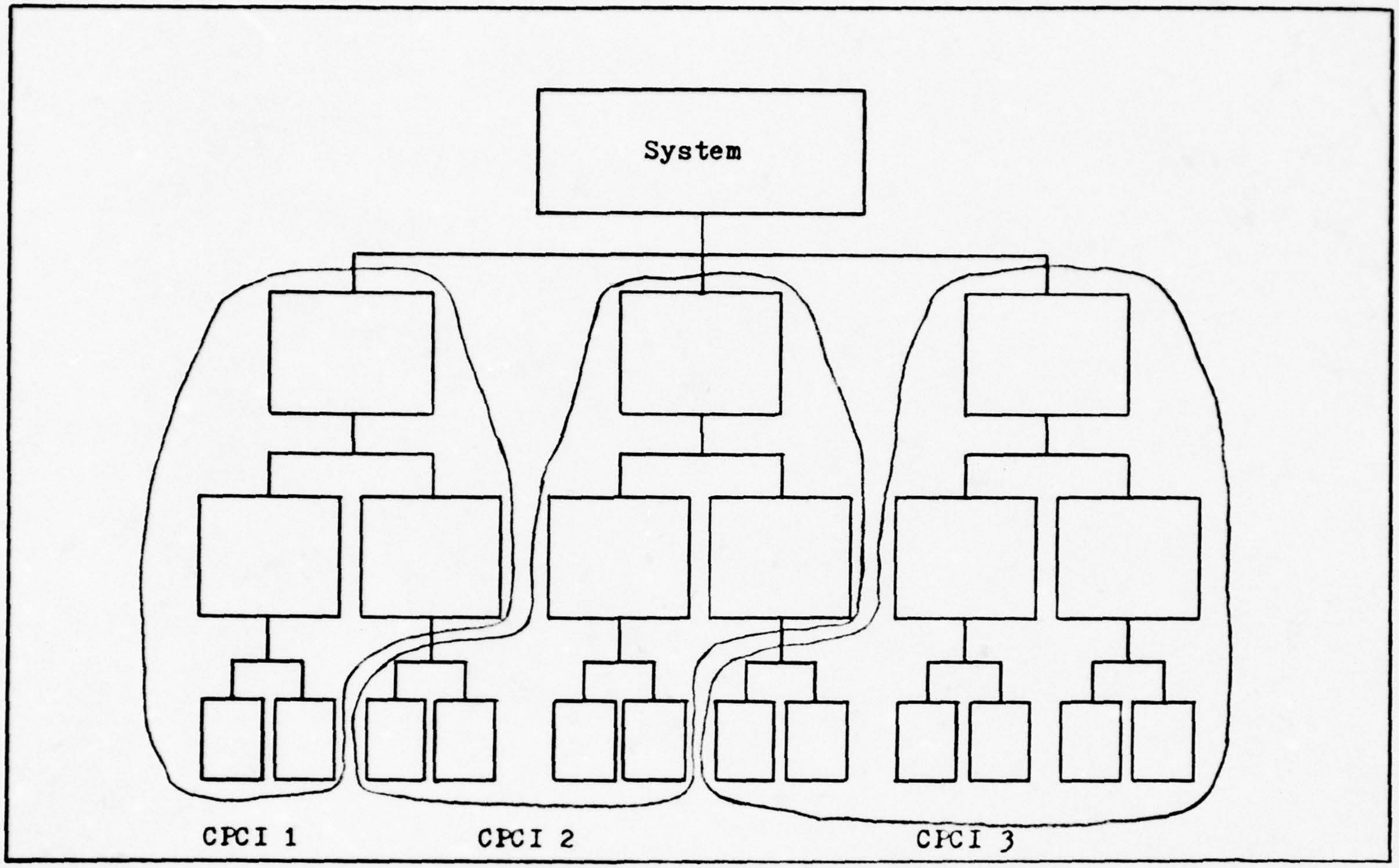


Figure 9. Top-down Structured Design with CPCIs Selected by Horizontal Allocation

VI. Summary, Conclusions, and Recommendations

This thesis documents a study of the software requirements allocation process in the acquisition and management of a major defense system. Although the literature is replete with studies, technical papers, and conference proceedings that discuss the problems associated with software engineering and software acquisition management, none of them have directly addressed the CPCI selection process. Therefore, this study was welcomed by practically all of the Government and industry personnel contacted during the research period. For example, one high-level Government expert made the following observation, "If architects were to design buildings the way software engineers design software, one woodpecker could destroy civilization." In another case, an industry expert stated, "The present practice of selecting CPCIs is strictly a stab in the dark, and it's about time someone did some research on what has been black magic for too many years."

The overall objective of this study was to determine how the system requirements to be implemented via software are allocated to CPCIs, and to investigate the feasibility and potential impacts of an alternate methodology. An extensive literature search and a survey of Government and industry "software experts" was used to accomplish this objective. First, the DoD and AF policies and procedures on

CPCI selection were determined through the literature search. Then, a semi-structured interview questionnaire was designed to capture both objective and subjective data on the CPCI selection process currently used by SPOs and contractors. In addition, the questionnaire was used to collect the experts' opinions on the utility of horizontal allocation. The results were most encouraging and should lead to a better understanding of this critical design issue. In addition to a summary of these results, some conclusions and recommendations are presented in the following paragraphs.

Summary of Results

DoD and AF Policies and Procedures for CPCI Selection. The acquisition and management of computer resources embedded in major defense systems is controlled by a complex hierarchy of policies and directives. Unfortunately, only a few of the documents address the process of selecting CPCIs directly. On the other hand, several do present guidelines and considerations for applying configuration management concepts to the acquisition and management of software.

In summary, the DoD and AF policies on CPCI selection state, "Computer resources will be managed as elements or subsystems of major importance during the system acquisition life cycle." Furthermore, "Computer hardware and software must be specified and treated as configuration items." These policies have led to the development of several instructions, regulations, pamphlets, and standards. Although these documents in general establish uniform policies and procedures for the implementation of configuration management within the DoD, the actual application to software is usually a management decision. Thus, several considerations for use in allocating system requirements to CPCIs are presented in these regulatory publications. They are as follows:

a. The initial CI list should contain more items than is anticipated on the final list.

b. Each CI should be produced and tested by a single contractor as an entity.

c. Processes that interact strongly should be assigned to the same CPCI.

d. Processes that will execute in different computers should be allocated to different CPCIs.

e. Processes whose development can feasibly be finished at significantly different times should be assigned to different CPCIs.

f. Include in each CPCI no more than an individual Government monitor can efficiently track.

g. The life cycle cost and management impacts associated with selecting CPCIs should be considered since choosing too few or too many can adversely affect the program.

h. System trade-offs and the natural decomposition of the software should be considered.

These policies and considerations have been interpreted in a few studies, technical reports, conference proceedings, lessons learned, and guidebooks. Thus, several additional criteria and/or factors to be considered in selecting CPCIs were described in Chapter III of this thesis. They are summarized as follows:

a. Each operational program should be a CPCI.

b. Premature "freezing" of system structure into management structure results in efforts being duplicated, system integration becoming more difficult, and overall visibility being reduced as the system design evolves.

c. Software tools should be separate CPCIs; however, resident debug programs are part of the operational CPCI.

d. Decompose the system so that management control and flexibility can be maintained.

e. Partition the system so that implementation, debugging, and testing of modules or groups of modules (CPCIs) can proceed in parallel.

f. CPCIs should be identified on the basis of whether they satisfy an end-use function, are key programs, have expected interface problems, and will provide added visibility to a particular computer program.

g. The CPCI selection decision should be made early in the program.

h. Include the AFLC system manager in the selection process.

i. The number of CPCIs should be minimized.

j. The end result of choosing a CPCI as a unit of management should be the ability to directly observe that a meaningful portion of the system works. Thus, a structure in which the functional flow across the system is divided into several CPCIs is artificial.

These interpretations have resulted in some discrepancies between the regulations and MIL-STDs. For example, the ASPR and AFSCP 800-3 consider computer software to be an item of data in the same manner as reports, forms, manuals, and specifications are. However, the Air Force policy, as stated in AFSCRP 800-1, is that computer software must be subjected to the same contractual and cost controls as hardware.

Similarly, AFR 800-14 and MIL-STD-483 do not agree on the level of detail required in the Part I specification for software.

Selection Criteria Currently in Use. Over 70% of those "software experts" participating in the survey indicated that they were familiar with the criteria used for allocating system requirements to CPCIs on their program. Those most often cited by the respondents were as follows:

a. Software developed by separate contractors/vendors must be separate CPCIs.

b. Functional modularity (i.e., grouping like functions) is a necessity.

c. On-line and off-line diagnostics and other support software must be separate CPCIs.

d. Each CPCI must operate on a single computer.

e. The number of CPCIs and interfaces must be minimized.

f. Keep the complex interfaces within a CPCI.

g. Off-the-shelf and developmental software must be separate CPCIs.

h. Separate locations and functions require separate CPCIs.

In addition, some other considerations were identified by the respondents. They included the following:

a. It should be possible to effectively test a CPCI as a whole.

b. Separate users/maintainers should signal separate CPCIs.

c. System performance parameters, such as MADT, Pt, should be considered in the allocation process.

d. "Politics" among the contractors, procuring agency, using organization, and maintenance organization must be considered.

e. Previous experience with a similar type of system is desirable.

f. Documentation requirements should be minimized.

g. User and maintainer advocacy should be considered.

h. Size is no longer considered a criterion to be used in selecting CPCIs.

Advantages/Disadvantages of These Criteria. Several advantages and disadvantages of the way CPCIs are presently selected were listed by the respondents. They were categorized by the method of their selection (i.e., contractor chosen, government imposed, previous experience of the contractor and/or SPO, and conformance with the DoD and AF policies and procedures).

In general, there was agreement that those system requirements allocated by the contractors result in a total collection of system software that can be managed in accordance with established Air Force procedures. In summary, the respondents described the following advantages of contractor-selected CPCIs:

a. The Government can hold the contractor responsible for a totally integrated design package, which is available at acceptance.

b. Cost to the Government is minimized.

c. The Government does not accept responsibility prematurely.

d. Programmer assignment is easier.

e. Unit testing and documenting the system are easier.

On the other hand, the proper application of CPCI selection criteria sometimes requires more knowledge and experience in system acquisition concepts and implications than is brought to bear by the contractors.

At the other extreme (i.e., government-imposed selection criteria/CPCI structure), the disadvantages appear to greatly outweigh the advantages. In other words, only three minor advantages were mentioned by the respondents; while, several negative results were directly attributed to the Government direction. For example, several costly ECPs were required on one program. In another case, completion of the functional qualification test did not allow observation of a working portion of the system.

Although most of the respondents cited previous experience as the primary consideration in selecting a CPCI structure, they did not agree on its utility. For example, experience may lead to an unnatural structure because "it was done this way before or the people in charge are unaware of any better technique." Similarly, selection criteria chosen on the basis of experience are hard to clarify and often overlap. This results in independent systems, duplicated functions, and difficult integration problems. On the other hand, criteria chosen in this manner do minimize the problems associated with specification maintenance, testing/requirements traceability, and turnover.

Only 2 of the 37 respondents indicated that the CPCI selection criteria on their program were chosen to conform with the DoD and AF policies and procedures. However, they did agree that the criteria were understandable and worked for most applications, but that all configuration contingencies may not have been considered.

Impacts of the Selected CPCI Structure. The number and composition of configuration items in the acquisition of a major defense system is a critical design issue since the Government's technical management activities primarily focuses on CIs and CPCIs. Thus, the need for a CPCI structure that allows management to draw realistic conclusions about the cost, schedule, and performance of the final software system cannot be overemphasized.

Over 72% of those Government and industry "software experts" interviewed in this research reported that they had experienced problems with the CPCI structure on their program. In fact, this is sufficient statistical evidence to support the hypothesis that problems are experienced with the CPCI structure on more than half of the system development/acquisition programs that are software-intensive. Another interesting statistical inference is that problems with the CPCI structure become more prevalent as a program gets larger.

In many cases, the impact of these problems was devastating to the system development/acquisition program. For example, schedules were slipped, documentation became confusing, requirements were "lost," ECPs were generated to modify the initial CPCI list, users became frustrated, testing requirements were relaxed, costs increased dramatically, and CPCIs were delivered with a large number of software discrepancies. In one case, the impacts were so extreme that the program was cancelled in lieu of trying to salvage it.

Feasibility, Potential Impacts, and Implementation of Horizontal Allocation

Feasibility. Over 82% of those responding to the Software Requirements Decomposition Interview indicated that horizontal allocation is feasible. In other words, there is sufficient statistical evidence to support the hypothesis that more than half of the "software experts" in the Government and industry believe that horizontal allocation is a feasible methodology for allocating software requirements to CPCIs. This result was significantly correlated (positively) with the size of the development/acquisition program, both in terms of the number of CPCIs and lines of code. Similarly, those respondents who experienced problems with the CPCI structure on their program were more likely to agree that horizontal allocation is feasible.

Potential Impacts. Based on an analysis of the interview results, the potential impacts of defining CPCIs in terms of system versions or models (each of which contains end-use system functional capabilities) were overwhelmingly positive. In fact, there was sufficient statistical evidence to support the hypothesis that more than half of the

Government and contractor "software experts" believe that horizontal allocation would be at least somewhat effective in favorably impacting all but one of the twelve evaluation parameters. In addition, over 60% of the respondents indicated that this approach to CPCI selection would be moderately to very effective in:

a. providing the program manager and user with more objective visibility into the system development,

b. improving the software quality,

c. reducing the software integration task, and

d. highlighting problems earlier.

The bivariate correlations among the 12 evaluation parameters, and between these parameters and the demographics/ program characteristics produced several significant interdependencies. For example, the respondents' opinions on the effectiveness of horizontal allocation in making debugging and testing more efficient, making software maintenance easier and less costly, increasing the morale of those working on the program, reducing the complexity of the decomposition process, and improving training effectiveness were all positively correlated with the number of CPCIs on those programs evaluated.

Implementation. Over 75% of those participating in the survey indicated that horizontal allocation should be implemented on future software development/acquisition programs. This result was significantly correlated (positively) with the size of the program evaluated, both in terms of the number of CPCIs and lines of code. In other words, respondents who evaluated large development/acquisition programs believed that this approach should be implemented in the future.

Conclusions

The importance of computer hardware and software in the Department of Defense has increased over the past 20 years to the point where computer technology is vital to the defense of our country. In addition, this technology has placed a tremendous strain on the fiscal assets of the DoD and the AF. Therefore, the need to manage computer hardware and software as critical components of a defense system throughout its life cycle is becoming widely recognized. A general awareness of this need was reflected in the survey of Government and industry "software experts," as well as in the literature.

Although the selection of CPCIs is one of the most critical decisions made during the acquisition of a software-intensive system, no known and proven approach was uncovered in this research. On most programs evaluated, functional modularity and previous experience were the primary considerations used by SPOs and contractors to allocate system requirements into CPCIs. This resulted in many problems with assessing development status, achieving system integration, completing meaningful tests and, documenting the system. In essence, managing a program partitioned in this manner forces the program manager to make an inductive evaluation that the system will ultimately work if and only if all other CPCIs work.

On the other hand, selecting CPCIs on the basis of horizontal allocation promises to favorably impact the cost, schedule, performance, and management parameters normally associated with an acquisition program. This alternate technique is based on defining a software-intensive system in terms of system versions or models, each of which contains end-use system functional capabilities. Therefore, a CPCI is equated to a version/model or series of versions/models for management purposes. In other words, a CPCI is a functional path or sequence of activities that results in the production of a required response from a set of available inputs or stimuli.

This alternative technique for allocating system requirements to CPCIs allows a system to be developed on an incremental basis rather than the traditional "black-box" approach. In other words, incremental development provides a system capability that is operational, rather than pieces and parts of a system that are only partially operational. Thus, "builds" represent functional capabilities and become the management milestones of system development.

Recommendations

The recommendations that follow are intended to provide suggested ways of improving the process of developing/

acquiring software embedded in major defense systems.

a. A total systems approach to the CPCI selection process is needed. To do this, it is necessary to consider the following factors/questions prior to finalizing the CPCI structure:

   (1) Advantages and disadvantages of each alternative CPCI structure.

   (2) Roles and responsibilities of each participating organization (i.e., user, developer, maintainer, and procurer).

   (3) Level of management visibility required.

   (4) Previous experience with a similar type of system.

   (5) Installation location and functions to be performed.

   (6) Contractual and legal implications of a CPCI.

b. The confusion and conflict caused by the ASPR and AFSCP 800-3 treating computer software as an item of data, rather than an active system component, should be terminated by changing these directives.

c. The discrepancies in AFR 800-14 and MIL-STD-483 regarding definition of the allocated baseline need to be resolved.

d. Since size is no longer considered a criterion to be used in selecting CPCIs, it should be deleted from all lists of CPCI selection criteria (Ref. AFSCP 800-7, 1977:76 and Glore, 1977:27).

e. The AFR 65-3 definition of a CI (CPCI is undefined in the regulation) should be clarified by changing "end-use function" to "end-use system capability." The rationale for this is that the definition as written leads to "vertical" allocation of software requirements to CPCIs. This structure typifies the "black box" approach; and therefore, creates artificial interfaces.

f. HQ AFSC should fund the development of an acquisition management guidebook that addresses, in detail, the system requirements allocation process (for CIs and CPCIs). A comprehensive example, as well as the results of this research should be included. In the meantime, the results of this research should be published and made available to the software industry, as well as those in the Government (procurers, developers, users, and maintainers).

g. Horizontal allocation should be implemented on a medium to small sized program to establish empirical evidence that it is as effective as the "software experts" believe it would be.

h. Since the number of "software experts" interviewed was limited, the survey should be expanded to include a larger number of respondents and variety of programs. This would increase confidence in the results achieved and conclusions derived in this study.

Bibliography

AFR 65-3. *Configuration Management*. Washington: Department of the Air Force, July 1975.

AFR 800-2. *Acquisition Program Management*. Washington: Department of the Air Force, November 1977.

AFR 800-14, Vol. I. *Management of Computer Resources in Systems*. Washington: Department of the Air Force, September 1975.

AFR 800-14, Vol. II. *Acquisition and Support Procedures for Computer Resources in Systems*. Washington: Department of the Air Force, September 1975.

AFSCP 800-3. *A Guide to Program Management*. Washington: Air Force Systems Command, April 1976.

AFSCP 800-7. *Configuration Management*. Andrews AFB, Maryland: Air Force Systems Command, December 1977.

AFSCRP 800-1. "Computer Software Acquisition Procedures," *Systems Management Newsletter*, *12*:3. Andrews AFB, Maryland: Air Force Systems Command, June 1975.

AFSC Supplement 1 to AFR 65-3. *Configuration Management*. Andrews AFB, Maryland: Air Force Systems Command, July 1975.

Airborne System Software Acquisition Engineering Guidebook: Configuration Management. 30323-6010-TU-00. Redondo Beach, California: TRW Defense and Space Group, November 1978.

Air Force Studies Board. *Operational Software Management and Development for U.S. Air Force Computer Systems*. Washington: Committee on Operational Software Management and Development of the Air Force Studies Board, Assembly of Engineering, National Research Council, 1977.

Anway, Mark D. *Configuration Management for the Development of Computer Systems*. Study Project Report. Defense Systems Management College, Fort Belvoir, Virginia, November 1976.

Asch, A., *et al*. *DoD Weapon System Software Acquisition and Management Study, Vol. I: MITRE Findings and Recommendations*. MTR-6908, Vol. I. McClean, Virginia: MITRE Corporation, May 1975 a.

Asch, A., *et al*. *DoD Weapon System Software Acquisition and Management Study, Vol. II: Supporting Material*. MTR-6908, Vol. II. McClean, Virginia: MITRE Corporation, June 1975 b.

Basili, Victor R. and Albert J. Turner. "Iterative Enhancement: A Practical Technique for Software Development," *IEEE Transactions on Software Engineering*, *SE-1* (4): 390-396 (December 1975).

Belford, P.C., *et al*. "Specifications: A Key to Effective Software Development," *Proceedings of IEEE/ACM 2nd International Conference on Software Engineering*, 71-79 (October 1976).

Bobbish, Charles E. "Lessons Learned: A C^3 Story," *AFSC Computer Resource Newsletter (AFSCRP 800-2)*, (4):2-4 (October 1978).

Boehm, Barry W., *et al*. *CCIP-85, Information Processing/Data Automation Implications of Air Force Command and Control Requirements in the 1980's: Executive Summary* (Revised Edition). Los Angeles, California: Space and Missile Systems Division, Air Force Systems Command, February 1972. (AD-742-292).

Brooks, Frederick P., Jr. *The Mythical Man-Month: Essays on Software Engineering*. Reading, Massachusetts: Addison-Wesley Publishing Co., 1974.

Carlson, William E. "Software Research in the DoD," *IEEE Transactions on Software Engineering*. 379-383 (October 1976).

Carlson, William E. and Barry C. DeRoze. *Defense System Software FY79-83 Research and Development Technology Plan*. Washington: Office of the Director of Defense Research and Engineering, September 1977. (AD-A047-062).

Caudill, Ray. *Applying Program Management Concepts to Software Development: An AFR 300-15 Critique*. Unpublished Study Project Report. Defense Systems Management College, Fort Belvoir, Virginia, October 1977. (AD-A052-126).

Cooper, Virgil L. "An Investigation of the Interview Method for Collecting Research Data," Unpublished paper written for SM 6.46, Organizational Behavior. School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, March 1979.

Davis, Ruth M. "Reducing Software Management Risks," *Defense Systems Management Review*, 1 (6):16-23 (Summer, 1978). (AD-A061-247).

DeRoze, Barry C. Embedded Computer Resources and the DSARC Process-A Guidebook. Washington: Office of Secretary of Defense, 1977. (AD-A046-398).

DeRoze, Barry C. and Thomas H. Nyman. "The Software Life Cycle-A Management and Technological Challenge in the DoD," IEEE Transactions on Software Engineering, SE-4 (4): 309-318 (July 1978).

Doane, Robert B. "The Evolving Nature of the C^3 Systems Acquisition Process," Unpublished white paper. Deputy for Technical Operations, Electronic Systems Division, Hanscom AFB, Massachusetts, April 1979.

DoDD 5000.1. Major System Acquisition. Washington: Department of Defense, January 1977.

DoDD 5000.2. Major System Acquisition Process. Washington: Department of Defense, January 1977.

DoDD 5000.29. Management of Computer Resources in Major Defense Systems. Washington: Department of Defense, April 1976.

Drezner, Stephen M., et al. The Computer Resource Management Study: Executive Summary. R-1855-PR. Santa Monica, California: Rand Corporation, September 1975.

Drezner, Stephen M., et al. The Computer Resources Management Study. R-1855/1-PR. Santa Monica, California: Rand Corporation, April 1976.

Farnan, David N. Reliable Computer Software: What It Is and How to Get It. Unpublished Study Project Report. Defense Systems Management College, Fort Belvoir, Virginia, November 1975. (AD-A026-988).

Glore, John B. Software Acquisition Management Guidebook: Life Cycle Events. ESD-TR-77-22. Bedford, Massachusetts: The MITRE Corporation, March 1977. (AD-A037-115).

Glore, John B., et al. Software Acquisition Management Guidebook: Regulations, Specifications, and Standards (Revised). ESD-TR-78-178. Bedford, Massachusetts: The MITRE Corporation, November 1978. (AD-A016-401).

Ikezawa, Michael A., et al. Ballistic Missile Defense Reliable Software Study: Final Report. CSS-75022-R2850. San Pedro, California: Logicon, Inc., November 1975. (AD-B008-031).

Kossiakoff, A., et al. DoD Weapon System Software Management Study. Laurel, Maryland: The Johns Hopkins University Applied Physics Laboratory, June 1975. (AD-A022-160).

Kuo, Franklin F., et al. Final Report by Communications, Command, Control and Intelligence(C^3I) Software Management Committee. Washington: Office of Secretary of Defense, August 1977.

Lientz, B.P., E.B. Swanson, and G. Tompkins. Maintenance Questionnaire and List of Respondents. Graduate School of Management, University of California at Los Angeles, Los Angeles, California, November 1978. (AD-A034-038).

Manley, John H. and Myron Lipow. Findings and Recommendations of the Joint Logistics Commanders Software Reliability Work Group, Vol. II: Supporting Technical Information. AFSC-TR-75-05-Vol. II. Andrews AFB, Maryland: Headquarters Air Force Systems Command, November 1975. (AD-A018-882).

Marciniak, John J., Lt. Col. "Software Acquisition Within Air Force Systems Command-A Management Approach," Defense Systems Management Review, 1 (6):32-39 (Summer 1978). (AD-A061-247).

McNichols, Charles W. An Introduction to Applied Multivariate Data Analysis. Unpublished Course Notes. School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, 1978.

MIL-STD-483, Notice 2. Configuration Management Practices for System, Equipment, Munitions, and Computer Programs. Washington: Department of the Air Force, March 1979.

Model Statement of Work Task for Software Development. Hanscom AFB, Massachusetts: Electronic Systems Division, March 1979.

Munson, John B. Draft Report of the Panel on Software Acquisition/Development Standards. Santa Monica, California: Joint Logistics Commanders Joint Policy Coordinating Group on Computer Resource Management-Software Workshop, Monterey, California during 3-5 April 1979, April 1979.

Myers, Ware. "The Need for Software Engineering," Computer, 2:12-25 (February 1978).

Neil, George. System Acquisition Management Guidebook: Reviews and Audits. ESD-TR-78-117. Santa Monica, California: System Development Corporation, November 19777. (AD-A052-567).

Nie, Norman H., et al. Statistical Package for the Social Sciences (Second Edition). New York: McGraw-Hill Book Co., 1975.

Nunnally, Jum C. Psychometric Theory. New York: McGraw-Hill Book Co., 1967.

Parnas, D.L. "On the Criteria to be Used in Decomposing Systems into Modules," Communications of the ACM, 15 (12): 1053-1058 (December 1972).

Ramamsorthy, C.V. "Testing Large Software with Automated Software Evaluation Systems," IEEE Transactions on Software Engineering, SE-1:46-58 (March 1975).

Runkle, Marty T., Capt. and Capt. Michael L. Smith. System Acquisition Guide. Maxwell AFB, Alabama: Air Command and Staff College, Air University, May 1978.

Salter, Kenneth G. "A Methodology for Decomposing System Requirements into Data Processing Requirements," Proceedings of IEEE/ACM 2nd International Conference on Software Engineering, 91-101 (October 1976).

Schoeffel, W.L. An Air Force Guide to Software Documentation Requirements. ESD-TR-76-159. Bedford, Massachusetts: The MITRE Corporation, June 1976. (AD-A027-051).

Searle, Lloyd V. AFAL Avionics Software Training Program Student Workbook for Course B: Software Acquisition and Management. TM-5365/103100. Santa Monica, California: System Development Corporation, October 1974.

-----. An Air Force Guide to Computer Program Configuration Management. ESD-TR-77-254. Santa Monica, California: System Development Corporation, August 1977. (AD-A047-308).

Self, Donnie B. A Study of the Acquisition and Management of Small-Scale Automatic Data Processing Equipment. Unpublished MS Thesis. School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, September 1978.

Selltiz, Claire, et al. Research Methods in Social Relations (Third Edition). New York: Holt, Rinehart, and Winston, Inc., 1976.

"Senate Report No. 938, October 22, 1965, to Accompany H.R. 4845 (Govern Operations Committee)," United States Code and Administrative News: 89th Congress--First Session 1965. Brooklyn: West Publishing Company and Edward Thompson Company, 1966.

Thurber, Kenneth J. "Requirements-Oriented Design: An Emerging Design Strategy," *Digest of Papers, 14th IEEE Computer Society International Conference (COMPCOM)*, 305-308 (Spring 1977).

Webster's New Collegiate Dictionary. Springfield, Massachusetts: G.&C. Merriam Co., 1977.

Weiss, David M. *The MUDD Report: A Case Study of Navy Software Development Practices*. Washington: Naval Research Laboratory, May 1975. (AD-A010-818).

Williams, R.D. *Managing the Development of Reliable Software*. TRW-SS-75-06. Redondo Beach, California: TRW Defense and Space System Group, April 1975.

Willmorth, N.E., *et al*. *Software Data Collection Study: Summary and Conclusions*. RADC-TR-76-329, Vol. 1 (of 8). Santa Monica, California: System Development Corporation, December 1976. (AD-A036-115).

Zabriskie, Ronald J. *Development of Weapons Systems Computer Programs: A Method for Control During FSD*. Unpublished Study Project Report. Defense Systems Management College, Fort Belvoir, Virginia, November 1975.

Zelkowitz, Marvin V. "Perspectives on Software Engineering," *Computing Surveys*, *10* (2):197-216 (June 1978).

APPENDIX A

Glossary of Terms

GLOSSARY OF TERMS

Acquisition. The act of gaining something through the conceptual, validation, full-scale development, production and deployment phases of the system life cycle.

Acquisition Life Cycle. The path, divided into phases, through which a program progresses between initial identification/documentation of an operational requirement and finally retirement from the operational retirement (AFSCP 800-7, 1977:84).

Allocation of Requirements. The act of apportioning the system performance and functional requirements to individually-identified subsets for purposes of managing their development. The identification of CPCIs is the typical result of this apportioning process.

Baseline. The documented requirements of a system that are implemented through multilateral agreements among a developer, procurer, and user (AFSCP 800-3, 1976:9-3).

Build. A collection of partial or complete program modules and data which, operating together, performs selected functions.

Computer Equipment (or Computer Hardware). Devices capable of accepting and storing computer data, executing a systematic sequence of operations on computer data or producing control outputs (DoDD 5000.29, 1976:Encl 1).

Computer Program. A series of instructions or statements in a form acceptable to computer equipment, designed to cause the execution of an operation or series of operations. Computer programs include such items as operating systems, assemblers, compilers, interpreters, data management systems, utility programs, maintenance/diagnostic programs and applications programs (DoDD 5000.29, 1976:Encl 1).

Computer Resources. The totality of computer equipment, computer programs, computer data, associated documentation, personnel, and supplies. These resources must be managed as elements or subsystems of major importance during all phases of the system acquisition life cycle (DoD 5000.29, 1976:Encl 1).

Computer Software. A combination of associated computer programs and computer data required to enable the computer equipment to perform computational or control function (DoDD 5000.29, 1976:Encl 1).

Configuration Identification (or CI/CPCI Selection). The process of documenting the performance, qualification, fabrication, and acceptance requirements of a system (AFSCP 800-7, 1977:16).

Configuration Item. An aggregation of hardware/computer programs or any of its discrete portions, which satisfies an end-use function and is designated by the Government for configuration management (AFR 65-3, 1974:A-2).

Configuration Management. The management discipline whose purpose is to identify, control, account for, and audit the functional and physical characteristics of systems, equipments, and other designated items developed, produced, operated, and supported by DoD components (AFR 65-3, 1975: 1-1).

Data. In this thesis, data refers to reports, forms, manuals, specifications, and other items of the classes which are acquired via a Contract Data Requirements List (Searle, 1977:111).

Decomposition of Requirements. The act of hierarchially separating system or subsystem requirements into units of development, such as functions, tasks, modules, subroutines, components, and instructions.

Embedded Computer Systems. A special subset of computer applications that are physically incorporated into a larger system. Their primary function is not data processing and their outputs generally include control signals.

Functional Modularity. The process of partitioning a system into modules, or standardized units, on the basis of grouping like functions to be performed by the system.

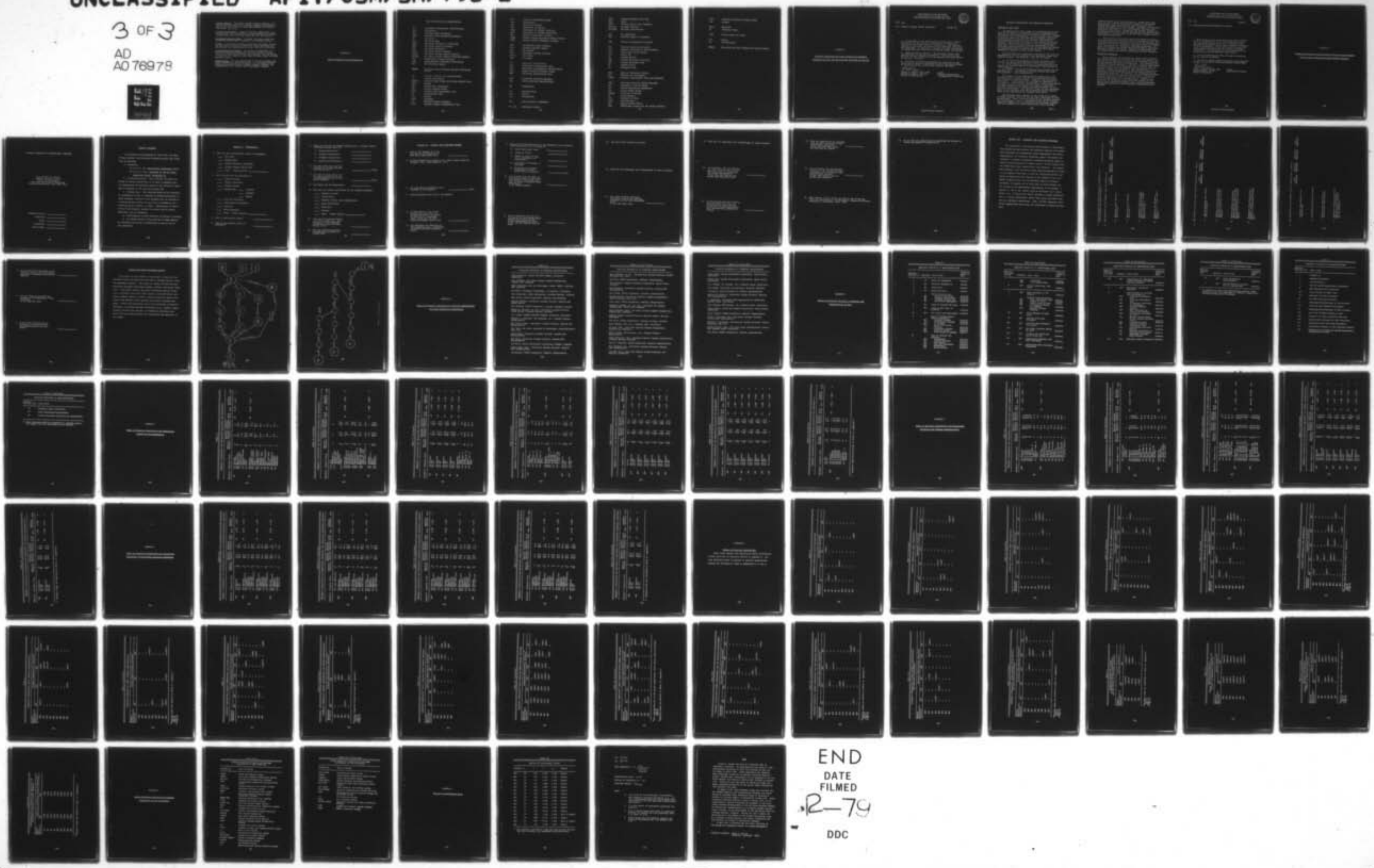
Horizontal Allocation. An approach to allocating system requirements to CPCIs on the basis of system versions or models, each of which contains end-use system functional capabilities. A system defined in this manner would be developed on an incremental basis in which each "build" provides a system capability that is operational, rather than pieces and parts that are only partially operational.

Major System Acquisition. A designation made by the Secretary of Defense. System programs involving an anticipated cost of $75 million in research, development, test and evaluation or $300 million in production are considered for designation as major system acquisitions (DoDD 5000.1, 1977: 2).

AD-A076 978 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOO--ETC F/G 15/5
AN INVESTIGATION OF THE SOFTWARE REQUIREMENTS ALLOCATION PROCES--ETC(U)
SEP 79 V L COOPER
UNCLASSIFIED AFIT/GSM/SM/79S-2 NL

3 OF 3

AD
A076978



END
DATE
FILMED
12-79
DDC

Program Manager. The person tasked to plan, organize, and conduct the acquisition of a system, subsystem or component within approved limits of system performance, and funding (AFR 800-2, 1977:1-2).

Software Engineering. Science of design, development, implementation, test, evaluation, and maintenance of computer software over its life cycle (DoDD 5000.29, 1976:Encl 1).

Software-Intensive System. A system that uses a high concentration of software to perform its intended function.

System. A composite of items, assemblies (or sets), skills, and techniques capable of performing and/or supporting an operational (or non-operational role) (AFR 65-3, 1974:A-3).

System Version (or Model). The actual configuration of a computer program configuration item(CPCI) which is introduced for installation and test or operation into the system in the form of a magnetic tape, deck of cards, disc, or other physical medium.

Weapon System. Any DoD system that is not of a general purpose, commercially available nature. It includes aircraft systems; space and missile systems; command, control, and communications(C^3) systems; and intelligence systems.

APPENDIX B

List of Acronyms and Abbreviations

LIST OF ACRONYMS AND ABBREVIATIONS

ACI	Allocated Configuration Identification
Acq.	Acquisition
ADP	Automatic Data Processing
ADPE	Automatic Data Processing Equipment
AF	Air Force
AFB	Air Force Base
AFIT	Air Force Institute of Technology
AFLC	Air Force Logistics Command
AFR	Air Force Regulation
AFSC	Air Force Systems Command
AFSCP	Air Force Systems Command Pamphlet
AFSCRP	Air Force Systems Command Recurring Pamphlet
ASD	Aeronautical Systems Division
ASPR	Armed Services Procurement Regulations
ATC	Air Training Command
BMDATC	Ballistic Missile Defense Advanced Technology Center
C^3	Command, Control, and Communications
C&C	Command and Control
CADSAT	Computer Aided Design and System Analysis Tool
Capt.	Captain
CDC	Control Data Corporation
CDR	Critical Design Review
CDRL	Contract Data Requirements List
CI	Configuration Item
Col.	Colonel
Cont.	Continued
CPC	Computer Program Component
CPCI	Computer Program Configuration Item

DCP	Decision Coordinating Paper
Dev.	Development
d.f.	Degrees of Freedom
DoD	Department of Defense
DoDD	Department of Defense Directive
DoDI	Department of Defense Instruction
DPC	Defense Procurement Circular
DSARC	Defense System Acquisition Review Council
DSMC	Defense Systems Management College
ECP	Engineering Change Proposal
ECS	Embedded Computer System
e.g.	for example
ESD	Electronic Systems Division
et al	and others
etc.	et cetera
F^3	Form, Fit, and Function
FCA	Functional Configuration Audit
FCI	Functional Configuration Identification
FCRC	Federal Contract Research Center
FQT	Functional Qualification Test
GFE	Government Furnished Equipment
GSA	General Services Administration
HQ	Headquarters
ID	Identification
i.e.	that is
Inc.	Incorporated
JLC	Joint Logistics Commanders
Lt. Col.	Lieutenant Colonel

MADT	Maximum Allowable Down-time
Maj.	Major
MENS	Mission Element Need Statement
MIL-STD	Military Standard
MIL-SPEC	Military Specification
N/A	Not Applicable
NBS	National Bureau of Standards
OMB	Office of Management and Budget
PCA	Physical Configuration Audit
PCI	Product Configuration Identification
PDR	Preliminary Design Review
P.L.	Public Law
PM	Program Manager
PMD	Program Management Directive
PMP	Program Management Plan
PO	Program Office
Pt	Processing Time
RADC	Rome Air Development Center
R&D	Research and Development
RDT&E	Research, Development, Test and Evaluation
SADT	Structured Analysis Design Technique
SAF	Secretary of the Air Force
SDC	System Development Corporation
SDR	System Design Review
SECDEF	Secretary of Defense
SM	System Manager
SOW	Statement of Work
SPECS	Specifications
SPO	System Program Office
SPSS	Statistical Package for the Social Sciences

SRWG	Software Reliability Working Group
SYS	Systems
Tech.	Technical
TR	Technical Report
USAF	United States Air Force
Vol.	Volume
VP	Vice-President
WWMCCS	World Wide Military Command and Control System

APPENDIX C

Interview Questionnaire Cover Letters for Telephone Interview (10 May 79) and Personal Interview (4 Jun 79)

DEPARTMENT OF THE AIR FORCE
AIR FORCE INSTITUTE OF TECHNOLOGY (ATC)
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433

REPLY TO
ATTN OF: ENA

SUBJECT: Master's Degree Thesis Interview 10 May 79

TO:

1. As promised in our recent phone conversation, a brief description of the Software Requirements Decomposition Interview is enclosed for your information. The specific purpose of the study and format of the interview instrument are described. In addition, a request for your cooperation in completing this vital data collection effort is included.

2. Based on pretest analyses, the interview should take approximately 45 minutes to complete. The specific date and time of the interview will be arranged with your office in the next 3-4 weeks.

3. Questions or comments regarding the interview or the thesis topic should be addressed to me at the above address or telephone number 513-255-5533 (Autovon 785-5533).

Virgil L Cooper

Virgil L. Cooper, Capt, USAF
Graduate Student, GSM-79S
School of Engineering

1 Atch
Software Requirements
Decomposition Interview

SOFTWARE REQUIREMENTS DECOMPOSITION INTERVIEW

Purpose of the Study

The objective of this study is to investigate one portion of the process of decomposing system performance requirements into individually-identified subsets for purposes of managing their development. These subsets, called configuration items(CIs) for hardware and computer program configuration items(CPCIs) for software, are normally identified during the early stages of a system acquisition program. This process is guided by the Air Force policy that there must be a recognized and documented initial statement of requirements, and that once stated, any change in requirements will be documented so that the current status of a program is known (AFSCP 800-7).

CIs and CPCIs are, therefore, regarded as the level of management at which a program office exercises formal management control. Specifically, they are the level:

a. At which the procuring activity specifies, contracts for, and accepts individual parts of the system.
b. Below which the developer is responsible for management of the development or procurement, and assembly of item components.
c. Above which the procuring activity retains the responsibility for delivering a complete system that meets the specified system requirements.

The portion of the decomposition process that is of concern in this research effort is the selection of CPCIs on a software-intensive system. Normally, the performance and functional requirements of a system are documented in a system or system segment specification. These requirements are then allocated to CIs and CPCIs based on selection criteria defined by the procuring activity and/or the development contractor. The specific topic of this interview, then, is the criteria used for decomposing system requirements to be implemented via software into CPCIs.

The research data gathered in this study will support a master's degree thesis at the Air Force Institute of Technology(AFIT). This topic was chosen for my research project for two reasons. First, I experienced many of the frustrations and disappointments associated with software acquisition as a member of a major command and control systems program office in my last assignment. During that four year period, I participated in all phases of the system acquisition cycle as a Computer Systems Analyst and a System Integration and Test Monitor. Secondly, my projected follow-on assignment is to a Headquarters Air Force Acquisition Study Group. Hopefully, the lessons learned in my previous assignment and the results of this research will contribute to improvements in the Air Force weapons system acquisition procedures.

You have been selected as part of a sample of government and contractor personnel who are presently(or have been in the recent past) responsible for one of the various disciplines of software development and acquisition management. Your cooperation is, therefore, sincerely requested. Answers or comments you provide may be included in published articles, reports, or texts. However, an individual and his organization will not be identified with his comments.

Interview Structure

This interview was designed to capture both objective and subjective data on previous CPCI selection efforts, as well as the respondents' opinions on an alternate technique of decomposing the system requirements. The questions are divided into three sections. Section I involves demographic questions, such as organizational level, type of job, experience level, size of project under development, etc. Section II contains questions on the criteria used for decomposing system requirements on recent and existing developmental projects. In addition, the advantages and disadvantages of the criteria are of interest. Section III involves questions on the feasibility and impact of an alternate technique for selecting CPCIs. Specifically, data will be collected on the perceived impacts on the system cost, system development time, system performance, management visibility, and contractor's management procedures.

DEPARTMENT OF THE AIR FORCE
AIR FORCE INSTITUTE OF TECHNOLOGY (ATC)
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433

REPLY TO ATTN OF: ENA

4 Jun 79

SUBJECT: Software Requirements Decomposition Interview

TO:

1. After completing some of the subject interviews at ASD and HQ AFLC, I concluded that some of the questions may require some considerable thought on your part. Therefore, I have enclosed the questionaire and an example referred to in Section II of the questionaire. If you have time prior to my arrival, please answer all questions that don't require any further clarification. This procedure will require less time for our meeting and will be more beneficial to my research.

2. I will be contacting your office this week to arrange the specific time for our meeting.

3. Questions or comments should be addressed to me or the above address or at commercial 513-236-2944 (Autovon 785-5533/3030).

Virgil L Cooper

VIRGIL L. COOPER, Captain, USAF
Graduate Student, GSM 79S
School of Engineering

1 Atch
Questionnaire w/example

APPENDIX D

Software Requirements Decomposition Interview Questionnaire

With a Simple Spacetrack System Example Attached

SOFTWARE REQUIREMENTS DECOMPOSITION INTERVIEW

Capt. Virgil Lee Cooper
Graduate Student, GSM-79S
School of Engineering
Air Force Institute of Technology
Wright-Patterson Air Force Base, Ohio

Interview Method ____________________

Date/Time ____________________

Place ____________________

Respondent ____________________

Phone Number ____________________

Privacy Statement

In accordance with paragraph 30, AFR 12-35, Air Force Privacy Program, the following information about this interview is provided:

a. Authority.

(1) 4 U.S.C. 301, Departmental Regulations and/or

(2) 10 U.S.C. 8012, Secretary of the Air Force, Powers and Duties, Delegation by.

b. Principal purposes. This interview is being conducted to collect information to be used in research aimed at illuminating and providing inputs to the solution of problems of interest to the Air Force and/or DOD.

c. Routine Uses. The interview data will be converted to information for use in research of system acquisition related problems. Results of the research will be included in a written master's thesis and may also be included in published articles, reports, or texts. Distribution of the results of the research, whether in written form or orally presented, will be unlimited.

d. Participation in this interview is entirely voluntary.

e. No adverse action of any kind may be taken against an individual who elects to participate in any or all of this interview.

Section I: Demographics

1. What is your organizational level of assignment?

_____ Air Staff

_____ MAJCOM Staff

_____ Product Division Staff(AFSC)

_____ Systems Program Office(SPO)

_____ Other. Please specify ____________________.

2. What function are you assigned to?

_____ Command Section

_____ Program Management

_____ Program Control

_____ Engineering: _____ Software

_____ Hardware

_____ System

_____ Test and Evaluation

_____ Configuration Management

_____ Logistics

_____ Data Management

_____ Other. Please specify ____________________.

3. What is your present grade? ____________________

4. What is your highest level of Education? ____________________

5. Have you ever had any formal training (e.g., College Courses, Technical Courses, etc) in:

   a. System Engineering? ____________________

   b. Software Engineering? ____________________

   c. Hardware Engineering? ____________________

   d. Acquisition Management? ____________________

6. How many years have you been involved in systems development/acquisition? ____________________ years

7. How many of those years have you been involved in the development/acquisition of computer software? ____________________ years

8. How recent was the experience? ____________________

9. What was the primary application of the computer software?

   _____ Airborne Systems

   _____ Intelligence

   _____ Command, Control, and Communication

   _____ Space and Missile

   _____ Scientific

   _____ Business

   _____ Other. Please specify ____________________.

10. What type of software experience do you have (e.g., programming, systems analysis, maintenance, test and evaluation, etc)? ____________________

11. How many system acquisition programs have you been involved with? ____________________

Section II: Present CPCI Selection Process

1. What is the Program ID of the program you are presently(or were last) associated with? ____________________

2. Briefly describe the program (e.g., what is(was) being developed? How? Major program? etc).

3. How long have you been(were you) assigned to the program? ____________________ years

4. Briefly describe your job on the program.

5. At what point in the system acquisition cycle were the system requirements allocated to CPCIs (e.g., conceptual phase, first part of full-scale development phase, etc)? ____________________

6. Who performed the functions of CPCI selection (e.g., contractor, Government software engineers, etc)? ____________________

7. What are the characteristics of the program you are presently (were) associated with in terms of:

   a. Total development cost? $________________

   b. Number of CPCIs? ________________

   c. Number of lines of code (i.e., executable instructions)? ________________

   d. Cost ratio of software to hardware? ________________

   e. Percentage of software development/contract period that is complete? ________________%

8. Do you agree that the CPCIs on your present(past) program were all defined at about the right size in terms of managing their development? ________________
   If no, please explain.

9. Are you familiar with the criteria used on your present(past) program for allocating the system requirements to CPCIs? ________________
   If so, can you identify some of them?

10. How were these criteria selected?

11. What are the advantages and disadvantages of these criteria?

12. Are these criteria different
from those used on any previous
programs you have been associated
with? ____________________
If so, what were they?

13. What are the advantages and disadvantages of those criteria?

14. In retrospect, can you identify
any other criteria that should
have been used in decomposing
the system requirements?
If so, what are they and why
should they have been used?

____

15. Do you believe that the criteria
used for selecting CPCIs are
different from those used for
decomposing system requirements
into configuration items(CIs)?
Explain.

____

16. Have you experienced any problems with the CPCI structure on your present(past) programs? If so, how were they solved and what was the impact? ____________________

17. Do you believe that an adequate analysis of the alternative CPCI structures was performed on your present(past) program? If not, what should have received more emphasis? ____________________

18. What changes, if any, would you like to see in the way CPCIs are selected (e.g., in philosophy, analysis techniques, regulations, procedures, etc)? Why?

19. Do you have any suggestions for improving the process of system requirements decomposition?

Section III: Alternate CPCI Selection Technique

The questions in this section are designed to investigate the feasibility and evaluate the potential impacts of an alternate technique for decomposing system requirements into CPCIs. Specifically, the alternate technique, called "horizontal allocation," is based on defining a software-intensive system in terms of system versions or models, each of which contains end-use system functional capabilities (See Example). For management purposes, each version or model would be defined as a CPCI. In its simplest terms then, a CPCI is a functional path or sequence of activities that results in the production of a required response from a set of available inputs or stimuli. Therefore, a complete set of CPCIs and CIs would satisfy all the functional and performance requirements of the system. A system defined in this manner would be developed on an incremental basis in which each "build" provides a system capability that is totally operational, rather than pieces and parts that are only partially operational. Thus, "builds" represent functional capabilities and become the milestones of system development.

1. Do you believe that this technique is feasible on a large software development project? ____________________

2. How effective do you believe that it would be in:

	Not Effective	Somewhat Effective	Moderately Effective	Very Effective
a. Reducing system development costs?				
b. Reducing system development time?				
c. Increasing objective visibility for project management, resulting in better project control?				
d. Improving the software quality (e.g., fewer latent errors)?				

2. How effective do you believe that it would be in: (cont.)

	Not Effective	Somewhat Effective	Moderately Effective	Very Effective
e. Making debugging and testing more efficient?				
f. Providing the user with more visibility into the development and performance of his system?				
g. Making software maintenance easier and less costly (e.g., less troubleshooting time and documentation easier to maintain)?				
h. Increasing the morale of programmers, coders, software engineers, and management personnel?				

2. How effective do you believe that it would be in: (cont.)

	Not Effective	Somewhat Effective	Moderately Effective	Very Effective
i. Reducing the software integration task?				
j. Highlighting problems earlier?				
k. Reducing the complexity of the decomposition process?				
l. Improving training effectiveness (e.g., by providing earlier hands-on training for the operator)?				

3. Do you think this technique would adversely impact the contractor's/developer's management procedures? ____________________
Explain.

4. Can you think of any other advantages or disadvantages of this technique? ____________________
If so, what are they?

5. Should this technique be implemented on future software development/acquisition programs? ____________________

Example of Simple Spacetrack System

The purpose of this system is to maintain a catalog of all man-made ofjects orbiting the earth and to provide specific status and management reports. The system is capable of receiving and processing messages from sensor radars, cameras, and visual sightings. Using these inputs, the system determines orbital parameters, correlates data with known objects, predicts future positions, detects changes in orbit, predicts satellite reentry and impact, and identifies and catalogs new objects launched into space. During this processing and at the conclusion of specific functions, the system provides information (e.g., alarms, visual displays or hard copy reports) to operational personnel upon which they will base decisions for controlling the operation of the system.

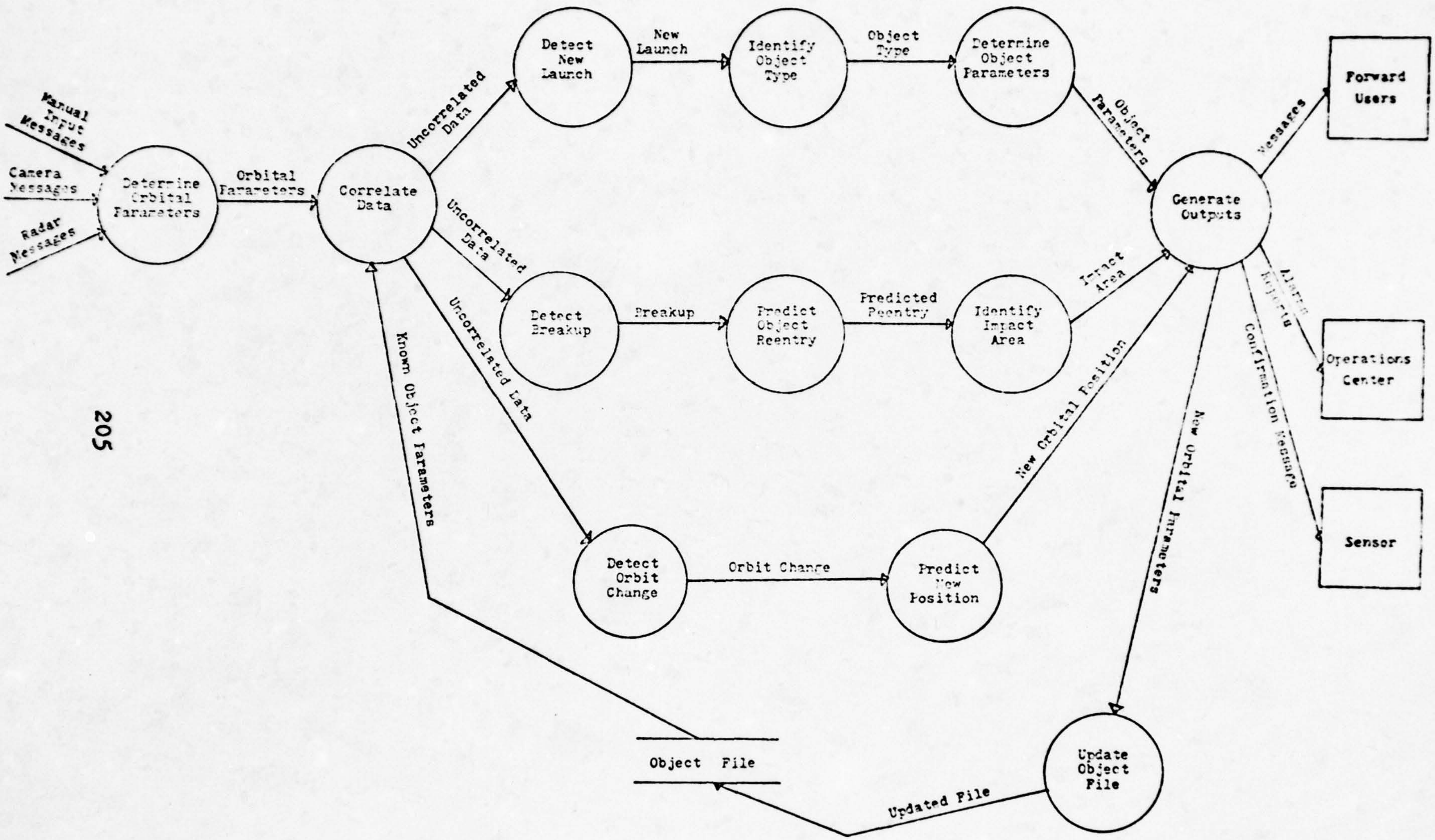


Figure [illegible] Data Flow Diagram (Functional Paths)

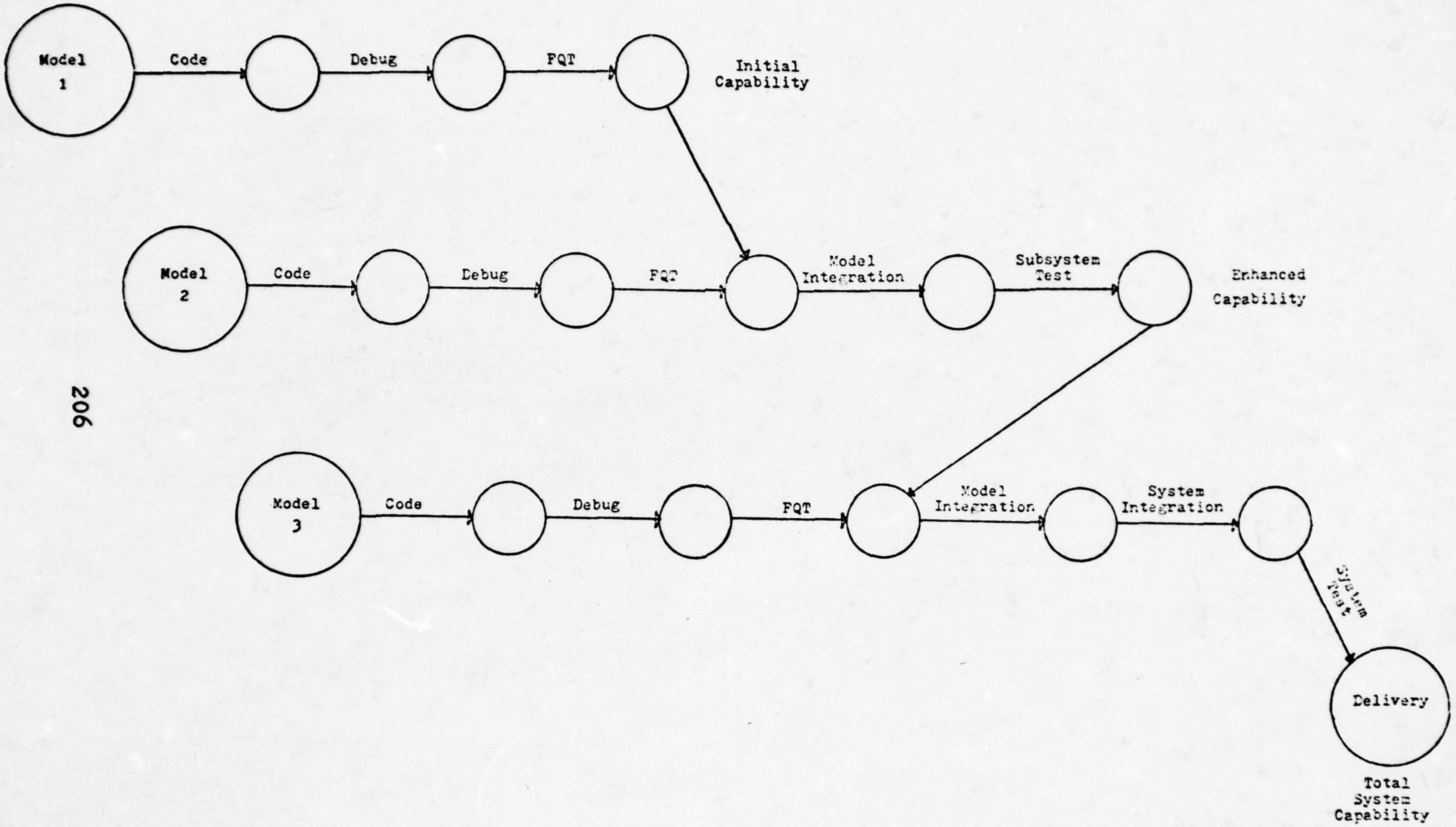


Figure 2. Model Development Cycle

APPENDIX E

Table of Personnel Responding to Interview Questionnaire and Their Respective Organization

TABLE III

PERSONNEL RESPONDING TO INTERVIEW QUESTIONNAIRE

Dale Albericci, Hughes Aircraft Company, Fullerton, California

Phil Anderson, Air Force Systems Command Headquarters, Andrews AFB, Maryland

Dean Bergstrom, Rome Air Development Center (AFSC), Griffiss AFB, New York

Bob Berri, Aerospace Corporation, Los Angeles, California

Bill Bjerstadt, MITRE Corporation, Colorado Springs, Colorado

Mike Bloom, MITRE Corporation, Bedford, Massachusetts

Charlie Bobbish, Electronic Systems Division, Hanscom AFB, Massachusetts

Dave Buckley, MITRE Corporation, Colorado Springs, Colorado

Howard E. Carolus, Lt. Col., Electronic Systems Division Field Office, Colorado Springs, Colorado

R.L. Clark, Hughes Aircraft Company, Fullerton, California

Vincent J. Cosentino, GTE Sylvania, Inc., Needham Heights, Massachusetts

Bob Cutter, Capt., Electronic Systems Division, Hanscom AFB, Massachusetts

Bill Dean, Air Force Institute of Technology, Wright-Patterson AFB, Ohio

Chuck Denny, Electronic Systems Division, Hanscom AFB, Massachusetts

Bob Doane, Electronic Systems Division, Hanscom AFB, Massachusetts

Bud Drutz, System Development Corporation, McLean, Virginia

Grace Dugas, Capt., Electronic Systems Division, Hanscom AFB, Massachusetts

Jim Earlain, MITRE Corporation, Bedford, Massachusetts

TABLE III (Continued)

PERSONNEL RESPONDING TO INTERVIEW QUESTIONNAIRE

Bill Fohrman, Lt. Col., Aeronautical Systems Division, Wright-Patterson AFB, Ohio

Bob Frye, MITRE Corporation, Bedford, Massachusetts

Kurt Fischer, Computer Sciences Corporation, Falls Church, Virginia

Sam Galzarano, Electronic Systems Division, Hanscom AFB, Massachusetts

Bob Gildea, MITRE Corporation, Bedford, Massachusetts

George Gehlauf, Air Force Logistics Command Headquarters, Wright-Patterson AFB, Ohio

John Glore, MITRE Corporation, Bedford, Massachusetts

Joseph S. Greene, Jr., Lt. Col., Strategic Air Command Headquarters, Offutt AFB, Nebraska

Dave Herrelko, Capt., Air Force Systems Command Headquarters, Andrews AFB, Maryland

Maretta Holden, Boeing Witcha Co./Seattle Branch, Seattle, Washington

Bob Irwin, MITRE Corporation, Colorado Springs, Colorado

Gary Klayman, TRW, Inc., Newberry Park, California

Al Kopp, Capt., Air Force Systems Command Headquarters, Andrews AFB, Maryland

Roger Linder, GTE Sylvania, Inc., Needham Heights, Massachusetts

Steve McCulloch, Maj., Aerospace Defense Command Headquarters, Colorado Springs, Colorado

David R. McMillan, MITRE Corporation, Bedford, Massachusetts

Norm Michaud, Col., Electronic Systems Division, Hanscom AFB, Massachusetts

Joe Mock, Maj., Space and Missile Systems Division, Los Angeles, California

TABLE III (Continued)

PERSONNEL RESPONDING TO INTERVIEW QUESTIONNAIRE

Jack Munson, System Development Corporation, Santa Monica, California

George Neil, System Development Corporation, Santa Monica, California

R.L. Oldham, TRW Systems, Inc., Rendondo Beach, California

Tony Pease, Booz-Allen and Hamilton, Bethesda, Maryland

Al Roberts, MITRE Corporation, Bedford, Massachusetts

Anthony D. Salvucci, Electronic Systems Division, Hanscom AFB, Massachusetts

C. Schneider, Air Force Plant Representative Office/TRW, Redondo Beach, California

Tony Schuman, TRW Systems, Inc., Redondo Beach, California

Lloyd Searle, System Development Corporation, Santa Monica, California

Oscar Shapiro, MITRE Corporation, Bedford, Massachusetts

Henry B. Stelling, Gen., Electronic Systems Division, Hanscom AFB, Massachusetts

Richard J. Sylvester, Aeronautical Systems Division, Wright-Patterson AFB, Ohio

George Trever, Capt., Air Force Plant Representative Office/TRW, Redondo Beach, California

Bill White, MITRE Corporation, Bedford, Massachusetts

APPENDIX F

Tables of Interview Questions, Variables, and Classification of Data

TABLE IV

QUESTIONS RESULTING IN QUANTIFIABLE DATA

Question (Section I)	Variable	Short Title	Classification of Data
1	Q1	Level of Assignment	Nominal
2	Q2	Function Assigned to	Nominal
3	Q3	Grade Level	Nominal
4	Q4	Education Level	Ordinal
5		Formal Training in:	
	Q5	System Engineering	Interval
	Q6	Software Engineering	Interval
	Q7	Hardware Engineering	Interval
	Q8	Acquisition Management	Interval
6	Q9	Years in Systems Dev./Acq.	Interval
7	Q10	Years in Dev./Acq. of Software	Interval
8	Q11	How recent was Experience?	Interval
9		Type of Software Application Experience:	
	Q12	Airborne Systems	Interval
	Q13	Intelligence Systems	Interval
	Q14	C^3 Systems	Interval
	Q15	Space and Missile Systems	Interval
	Q16	Scientific Systems	Interval
	Q17	Business System	Interval
	Q18	Other Types of Systems	Interval
10		Type of Software Experience:	
	Q19	Programming	Interval
	Q20	Systems Analysis	Interval
	Q21	Maintenance	Interval
	Q22	Test and Evaluation	Interval
	Q23	Engineering	Interval
	Q24	System Design	Interval

TABLE IV (Continued)

QUESTIONS RESULTING IN QUANTIFIABLE DATA

Question (Sections I & II)	Variable	Short Title	Classification of Data
	Q25	Management	Interval
	Q26	Other Experiences	Interval
11	Q27	Number System Acq. Programs	Interval
3	Q28	Years Assigned to the Program	Interval
7		Program Characteristics:	
	Q29	Total Development Cost	Ordinal
	Q30	Number of CPCIs	Ordinal
	Q31	Number Lines of Code	Ordinal
	Q32	Cost Ratio-Software to Hardware	Interval
	Q33	Percent Contract Complete	Interval
8*	Q34	CPCIs Defined at Right Size?	Interval
9*	Q35	Familiar with Past Criteria?	Interval
12*	Q36	Criteria Used Different from Past?	Interval
14*	Q37	Any Other Criteria Should Be Used?	Interval
15*	Q38	Criteria for CPCIs Different from CIs?	Interval
16*	Q39	Experienced Problems with CPCI Structure?	Interval
17*	Q40	Alternative CPCI Structure Evaluated?	Interval

TABLE IV (Continued)

QUESTIONS RESULTING IN QUANTIFIABLE DATA

Question (Sections II & III)	Variable	Short Title	Classification of Data
19*	Q41	Suggestions for Improving System Requirements Decomposition Process	Interval
1	Q42	Horizontal Allocation Feasible?	Interval
2		Effectiveness of Horizontal Allocation in:	
	Q43	Reducing Development Cost	Interval
	Q44	Reducing Development Time	Interval
	Q45	Increasing Visibility to Management	Interval
	Q46	Improve Software Quality	Interval
	Q47	More Efficient Debugging and Testing	Interval
	Q48	Increasing Visibility to User	Interval
	Q49	Making Software Maintenance Easier and Less Costly	Interval
	Q50	Increasing Morale	Interval
	Q51	Reducing Software Integration Task	Interval
	Q52	Highlighting Problems Earlier	Interval
	Q53	Reducing Complexity in Decomposition Process	Interval
	Q54	Improving Training Effectiveness	Interval
3*	Q55	Adversely Impact Contractor	Interval

TABLE IV (Continued)

QUESTIONS RESULTING IN QUANTIFIABLE DATA

Question (Section III)	Variable	Short Title	Classification of Data
4*	Q56	Other Advantages/ Disadvantages	Interval
5	Q57	Should Horizontal Allocation Be Implemented?	Interval

* In addition to soliciting quantifiable answers, these questions asked the respondent to explain, expand on or provide rationale for his answer.

TABLE V

QUESTIONS RESULTING IN QUALITATIVE DATA

Question (Section II)	Short Title
1	Program ID
2	Program Description
4	Job Description
5	Time of System Requirements Allocation
6	Who Performed CPCI Selection?
8*	CPCIs Defined at Right Size?
9*	Familiar with Past Criteria?
10	How Were Criteria Selected?
11	Advantages/Disadvantages of the Criteria
12*	Criteria Used Different From Past?
13	Advantages/Disadvantages of Past Criteria
14*	Any Other Criteria Should Be Used?
15*	Criteria for CPCIs Different From CIs?
16*	Experienced Problems with CPCI Structure?
17*	Alternative CPCI Structure Evaluated?
18	Recommended Changes in CPCI Selection Process
19*	Suggestions for Improving System Requirements Decomposition Process

TABLE V (Continued)

QUESTIONS RESULTING IN QUALITATIVE DATA

Question (Section III)	Short Title
3*	Adversely Impact Contractor
4*	Other Advantages/Disadvantages
5*	Should Horizontal Allocation Be Implemented?

* These questions asked the respondent to provide qualitative support for his dichotomous yes/no answers.

APPENDIX G

Table of Frequency Distribution and Descriptive Statistics for Demographics

TABLE VI

FREQUENCY DISTRIBUTION AND DESCRIPTIVE STATISTICS FOR DEMOGRAPHICS

Variable*	Response Group	Absolute Frequency	Relative Frequency (Percent)	Adjusted Frequency (Percent)	Valid Cases	Mean	Standard Deviation	Mode
Q1	(0) Missing	0	0	--	45	4.178	1.114	5
	(1) Air Staff	0	0	--				
	(2) MAJCOM Staff	6	13.3	13.3				
	(3) Product Division Staff	6	13.3	13.3				
	(4) Systems Program Office	7	15.6	15.6				
	(5) Other	26	57.8	57.8				
Q2	(0) Missing	1	2.2	--	44	6.037	4.037	4
	(1) Command Section	1	2.2	2.3				
	(2) Program Management	9	20.0	20.5				
	(3) Program Control	0	0	0				
	(4) Software Engineering	15	33.3	34.1				
	(5) Hardware Engineering	0	0	0				
	(6) System Engineering	4	8.9	9.1				
	(7) Test and Evaluation	0	0	0				
	(8) Configuration Management	1	2.2	2.3				
	(9) Logistics Management	0	0	0				

TABLE VI (Continued)

FREQUENCY DISTRIBUTION AND DESCRIPTIVE STATISTICS FOR DEMOGRAPHICS

Variable*	Response Group	Absolute Frequency	Relative Frequency (Percent)	Adjusted Frequency (Percent)	Valid Cases	Mean	Standard Deviation	Mode
	(10) Data Management	0	0	0				
	(11) Software Engineering and Configuration Management	3	6.7	6.8				
	(12) Other	11	24.4	25.0				
Q3	(0) Missing	5	11.1	--	40	6.200	1.897	8
	(1) General/GS-16	1	2.2	2.5				
	(2) Colonel/GS-15	0	0	--				
	(3) Lieutenant Colonel/GS-14	3	6.7	7.5				
	(4) Major/GS-13	3	6.7	7.5				
	(5) Captain/GS-12	10	22.2	25.0				
	(6) Lieutenant/GS-11 and GS-10	0	0	--				
	(7) MITRE Tech Staff	8	17.8	20.0				
	(8) Other (VP, Tech Director, Division Chief, etc.)	15	33.3	37.5				
Q4	(0) Missing	1	2.2	--	44	1.864	0.632	2
	(1) Bachelors and Below	12	26.6	27.3				
	(2) Masters	26	57.8	59.1				
	(3) Doctorate	6	13.3	13.6				

TABLE VI (Continued)

FREQUENCY DISTRIBUTION AND DESCRIPTIVE STATISTICS FOR DEMOGRAPHICS

Variable*	Response Group	Absolute Frequency	Relative Frequency (Percent)	Adjusted Frequency (Percent)	Valid Cases	Mean	Standard Deviation	Mode
Q5	(0) Missing	0	0	--	45	1.533	0.505	2
	(1) Yes	21	46.7	46.7				
	(2) No	24	53.3	53.3				
Q6	(0) Missing	1	2.2	--	44	1.545	0.504	2
	(1) Yes	20	44.4	45.5				
	(2) No	24	53.3	54.5				
Q7	(0) Missing	1	2.2	--	44	1.568	0.501	2
	(1) Yes	19	42.2	43.2				
	(2) No	25	55.6	56.8				
Q8	(0) Missing	1	2.2	--	44	1.568	0.501	2
	(1) Yes	19	42.2	43.2				
	(2) No	25	55.6	56.8				
Q9	(0) Missing	0	0	--	45	13.600	7.709	8
	(1) Less than 5 years	1	2.2	2.2				
	(2) 5 through 10 years	22	48.9	48.9				
	(3) 11 through 20 years	12	26.7	26.7				
	(4) More than 20 years	10	22.2	22.2				

TABLE VI (Continued)

FREQUENCY DISTRIBUTION AND DESCRIPTIVE STATISTICS FOR DEMOGRAPHICS

Variable*	Response Group	Absolute Frequency	Relative Frequency (Percent)	Adjusted Frequency (Percent)	Valid Cases	Mean	Standard Deviation	Mode
Q10	(0) Missing	1	2.2	--	44	12.341	7.091	10
	(1) Less than 5 years	3	6.7	6.8				
	(2) 5 through 10 years	22	48.9	50.0				
	(3) 11 through 20 years	11	24.4	25.0				
	(4) More than 20 years	8	17.8	18.2				
Q11	(0) Missing	0	0	--	45	1.244	0.802	1
	(1) Present	41	91.1	91.1				
	(2) Less than 1 year	0	0	--				
	(3) 1 through 3 years ago	1	2.2	2.2				
	(4) 3 through 5 years ago	3	6.7	6.7				
	(5) Greater than 5 years ago	0	0	--				
Q12	(0) Missing	0	0	--	45	1.733	0.447	2
	(1) Yes	12	26.7	26.7				
	(2) No	33	73.3	73.3				
Q13	(0) Missing	0	0	--	45	1.822	0.387	2
	(1) Yes	8	17.8	17.8				

TABLE VI (Continued)

FREQUENCY DISTRIBUTION AND DESCRIPTIVE STATISTICS FOR DEMOGRAPHICS

Variable*	Response Group	Absolute Frequency	Relative Frequency (Percent)	Adjusted Frequency (Percent)	Valid Cases	Mean	Standard Deviation	Mode
	(2) No	37	82.2	82.2				
Q14	(0) Missing	0	0	--	45	1.178	0.387	1
	(1) Yes	37	82.2	82.2				
	(2) No	8	17.8	17.8				
Q15	(0) Missing	0	0	--	45	1.756	0.435	2
	(1) Yes	11	24.4	24.4				
	(2) No	34	75.6	75.6				
Q16	(0) Missing	0	0	--	45	1.733	0.447	2
	(1) Yes	12	26.7	26.7				
	(2) No	33	73.3	73.3				
Q17	(0) Missing	0	0	--	45	1.822	0.387	2
	(1) Yes	8	17.8	17.8				
	(2) No	37	82.2	82.2				
Q18	(0) Missing	0	0	--	45	1.689	0.468	2
	(1) Yes	14	31.1	31.1				
	(2) No	31	68.9	68.9				
Q19	(0) Missing	0	0	--	45	1.222	0.420	1
	(1) Yes	35	77.8	77.8				
	(2) No	10	22.2	22.2				

TABLE VI (Continued)

FREQUENCY DISTRIBUTION AND DESCRIPTIVE STATISTICS FOR DEMOGRAPHICS

Variable*	Response Group	Absolute Frequency	Relative Frequency (Percent)	Adjusted Frequency (Percent)	Valid Cases	Mean	Standard Deviation	Mode
Q20	(0) Missing	0	0	--	45	1.222	0.420	1
	(1) Yes	35	77.8	77.8				
	(2) No	10	22.2	22.2				
Q21	(0) Missing	0	0	--	45	1.644	0.484	2
	(1) Yes	16	35.6	35.6				
	(2) No	29	64.4	64.4				
Q22	(0) Missing	0	0	--	45	1.422	0.499	1
	(1) Yes	26	57.8	57.8				
	(2) No	19	42.2	42.2				
Q23	(0) Missing	0	0	--	45	1.689	0.468	2
	(1) Yes	14	31.1	31.1				
	(2) No	31	68.9	68.9				
Q24	(0) Missing	0	0	--	45	1.644	0.484	2
	(1) Yes	16	35.6	35.6				
	(2) No	29	64.4	64.4				
Q25	(0) Missing	0	0	--	45	1.578	0.499	2
	(1) Yes	19	42.2	42.2				
	(2) No	26	57.8	57.8				
Q26	(0) Missing	0	0	--	45	1.911	0.288	2

TABLE VI (Continued)

FREQUENCY DISTRIBUTION AND DESCRIPTIVE STATISTICS FOR DEMOGRAPHICS

Variable*	Response Group	Absolute Frequency	Relative Frequency (Percent)	Adjusted Frequency (Percent)	Valid Cases	Mean	Standard Deviation	Mode
	(1) Yes	4	8.9	8.9				
	(2) No	41	91.1	91.1				
Q27	(0) Missing	1	2.2	--	44	6.591	7.851	4
	(1) 1 Program	5	11.1	11.4				
	(2) 2 Programs	4	8.9	9.1				
	(3) 3 Programs	5	11.1	11.4				
	(4) 4 Programs	11	24.4	25.0				
	(5) 5 Programs	3	6.7	6.8				
	(6) 6 to 10 Programs	10	22.2	22.7				
	(7) More than 10 Programs	6	13.3	13.7				

* Variable names are described in Table IV, Appendix F.

APPENDIX H

Table of Frequency Distribution and Descriptive Statistics for Program Characteristics

TABLE VII

FREQUENCY DISTRIBUTION AND DESCRIPTIVE STATISTICS FOR PROGRAM CHARACTERISTICS

Variable*	Response Group	Absolute Frequency	Relative Frequency (Percent)	Adjusted Frequency (Percent)	Valid Cases	Mean	Standard Deviation	Mode
Q28	(0) Missing	5	11.1	--	40	3.525	2.837	1
	(1) 1 year	11	24.4	27.5				
	(2) 2 years	9	20.0	22.5				
	(3) 3 years	6	13.3	15.0				
	(4) 4 years	2	4.4	5.0				
	(5) 5 years	4	8.9	10.0				
	(6) 6 through 10 years	7	15.6	17.5				
	(7) Greater than 10 years	1	2.2	2.5				
Q29	(0) Missing	12	26.7	--	33	4.424	1.696	6
	(1) Less than 1 million dollars	2	4.4	6.1				
	(2) 1 to 5 million dollars	5	11.1	15.2				
	(3) 5 to 25 million dollars	2	4.4	6.1				
	(4) 25 to 50 million dollars	5	11.1	15.2				
	(5) 50 to 100 million dollars	6	13.3	18.2				
	(6) 100 to 500 million dollars	13	28.9	39.4				
	(7) 500 million to 1 billion dollars	0	0	--				

TABLE VII (Continued)

FREQUENCY DISTRIBUTION AND DESCRIPTIVE STATISTICS FOR PROGRAM CHARACTERISTICS

Variable*	Response Group	Absolute Frequency	Relative Frequency (Percent)	Adjusted Frequency (Percent)	Valid Cases	Mean	Standard Deviation	Mode
	(8) Greater than 1 billion dollars	0	0	--				
Q30	(0) Missing	11	24.4	--	34	21.235	27.069	2
	(1) 1 CPCI	1	2.2	2.9				
	(2) 2 CPCIs	3	6.7	8.8				
	(3) 3 CPCIs	3	6.7	8.8				
	(4) 4 CPCIs	3	6.7	8.8				
	(5) 5 CPCIs	0	0	0				
	(6) 6 through 10 CPCIs	11	24.4	32.4				
	(7) 11 through 20 CPCIs	3	6.7	8.8				
	(8) 21 through 40 CPCIs	3	6.7	8.8				
	(9) 41 through 60 CPCIs	4	8.9	11.8				
	(10) Greater than 60 CPCIs	3	6.7	8.8				
Q31	(0) Missing	19	42.2	--	26	4.154	1.434	4
	(1) Less than 10,000 lines	2	4.4	7.7				
	(2) 10,000 through 50,000 lines	2	4.4	7.7				

TABLE VII (Continued)

FREQUENCY DISTRIBUTION AND DESCRIPTIVE STATISTICS FOR PROGRAM CHARACTERISTICS

Variable*	Response Group	Absolute Frequency	Relative Frequency (Percent)	Adjusted Frequency (Percent)	Valid Cases	Mean	Standard Deviation	Mode
	(3) 50,000 through 100,000 lines	1	2.2	3.8				
	(4) 100,000 through 500,000 lines	11	24.4	42.3				
	(5) 500,000 through 1,000,000 lines	5	11.1	19.2				
	(6) Greater than 1,000,000 lines	5	11.1	19.2				
Q32	(0) Missing	25	53.3	--	20	2.564	3.371	.250
	(1) Less than 1 to 1	9	20.0	45.0				
	(2) 1:1 through 2:1	4	8.9	20.0				
	(3) 2:1 through 3:1	2	4.4	10.0				
	(4) 3:1 through 4:1	1	2.2	5.0				
	(5) 4:1 through 5:1	1	2.2	5.0				
	(6) 5:1 through 10:1	2	4.4	10.0				
	(7) More than 10:1	1	2.2	5.0				
Q33	(0) Missing	17	37.8	--	28	69.000	34.658	100
	(1) Less than 20%	2	4.4	7.1				
	(2) 20 through 50%	10	22.2	35.7				
	(3) 51 through 80%	1	2.2	3.6				
	(4) 81 through 99%	7	15.6	25.0				
	(5) 100%	8	17.8	28.6				

TABLE VII (Continued)

FREQUENCY DISTRIBUTION AND DESCRIPTIVE STATISTICS FOR PROGRAM CHARACTERISTICS

Variable*	Response Group	Absolute Frequency	Relative Frequency (Percent)	Adjusted Frequency (Percent)	Valid Cases	Mean	Standard Deviation	Mode
Q34	(0) Missing	8	17.8	--	37	1.486	0.731	1
	(1) Yes	23	51.1	62.2				
	(2) No	11	24.4	29.7				
	(3) Undecided	2	4.4	5.4				
	(4) Unknown	1	2.2	2.7				
Q35	(0) Missing	8	17.8	--	37	1.297	0.463	1
	(1) Yes	26	57.8	70.3				
	(2) No	11	24.4	29.7				
Q36	(0) Missing	17	37.8	--	31	1.500	0.509	1
	(1) Yes	14	31.1	50.0				
	(2) No	14	31.1	50.0				
Q37	(0) Missing	14	31.1	--	31	1.387	0.495	1
	(1) Yes	19	42.2	61.3				
	(2) No	12	26.7	38.7				
Q38	(0) Missing	13	28.9	--	32	1.469	0.567	1
	(1) Yes	18	40.0	56.3				
	(2) No	13	28.9	40.6				
	(3) Somewhat	1	2.2	3.1				
Q39	(0) Missing	9	20.0	--	36	1.278	0.454	1
	(1) Yes	26	57.8	72.2				
	(2) No	10	22.2	27.8				

TABLE VII (Continued)

FREQUENCY DISTRIBUTION AND DESCRIPTIVE STATISTICS FOR PROGRAM CHARACTERISTICS

Variable*	Response Group	Absolute Frequency	Relative Frequency (Percent)	Adjusted Frequency (Percent)	Valid Cases	Mean	Standard Deviation	Mode
Q40	(0) Missing	9	20.0	--	36	1.611	0.494	2
	(1) Yes	14	31.1	38.9				
	(2) No	22	48.9	61.1				
Q41	(0) Missing	11	24.4	--	34	1.235	0.431	1
	(1) Yes	26	57.8	76.5				
	(2) No	8	17.8	23.5				

* Variable names are described in Table IV, Appendix F.

APPENDIX I

Table of Frequency Distribution and Descriptive Statistics for Horizontal Allocation Questions

TABLE VIII

FREQUENCY DISTRIBUTION AND DESCRIPTIVE STATISTICS FOR HORIZONTAL ALLOCATION QUESTIONS

Variable*	Response Group	Absolute Frequency	Relative Frequency (Percent)	Adjusted Frequency (Percent)	Valid Cases	Mean	Standard Deviation	Mode
Q42	(0) Missing	6	13.3	--	39	1.821	0.389	1
	(1) Yes	32	71.1	82.1				
	(2) No	7	15.6	17.9				
Q43	(0) Missing	7	15.6	--	38	2.421	1.130	2
	(1) Not Effective	11	24.4	28.9				
	(2) Somewhat Effective	8	17.8	21.1				
	(3) Moderately Effective	11	24.4	28.9				
	(4) Very Effective	8	17.8	21.1				
Q44	(0) Missing	7	15.6	--	38	2.474	1.156	1
	(1) Not Effective	11	24.4	28.9				
	(2) Somewhat Effective	7	15.6	18.4				
	(3) Moderately Effective	11	24.4	28.9				
	(4) Very Effective	9	20.0	23.7				
Q45	(0) Missing	7	15.6	--	38	3.000	1.139	4
	(1) Not Effective	7	15.6	18.4				
	(2) Somewhat Effective	3	6.7	7.9				
	(3) Moderately Effective	11	24.4	28.9				

TABLE VIII (Continued)

FREQUENCY DISTRIBUTION AND DESCRIPTIVE STATISTICS FOR HORIZONTAL ALLOCATION QUESTIONS

Variable*	Response Group	Absolute Frequency	Relative Frequency (Percent)	Adjusted Frequency (Percent)	Valid Cases	Mean	Standard Deviation	Mode
	(4) Very Effective	17	37.8	44.7				
Q46	(0) Missing	7	15.6	--	38	2.632	1.076	3
	(1) Not Effective	8	17.8	21.1				
	(2) Somewhat Effective	7	15.6	18.4				
	(3) Moderately Effective	14	31.1	36.8				
	(4) Very Effective	9	20.0	23.7				
Q47	(0) Missing	7	15.6	--	38	2.737	1.131	4
	(1) Not Effective	7	15.6	18.4				
	(2) Somewhat Effective	9	20.0	23.7				
	(3) Moderately Effective	9	20.0	23.7				
	(4) Very Effective	13	28.9	34.2				
Q48	(0) Missing	7	15.6	--	38	3.132	1.095	4
	(1) Not Effective	6	13.3	15.8				
	(2) Somewhat Effective	2	4.4	5.3				
	(3) Moderately Effective	11	24.4	28.9				
	(4) Very Effective	19	42.2	50.0				

TABLE VIII (Continued)

FREQUENCY DISTRIBUTION AND DESCRIPTIVE STATISTICS FOR HORIZONTAL ALLOCATION QUESTIONS

Variable*	Response Group	Absolute Frequency	Relative Frequency (Percent)	Adjusted Frequency (Percent)	Valid Cases	Mean	Standard Deviation	Mode
Q49	(0) Missing	8	17.8	--	37	2.297	1.051	3
	(1) Not Effective	11	24.4	29.7				
	(2) Somewhat Effective	9	20.0	24.3				
	(3) Moderately Effective	12	26.7	32.4				
	(4) Very Effective	5	11.1	13.5				
Q50	(0) Missing	8	17.8	--	37	2.405	1.166	1
	(1) Not Effective	12	26.7	32.4				
	(2) Somewhat Effective	6	13.3	16.2				
	(3) Moderately Effective	11	24.4	29.7				
	(4) Very Effective	8	17.8	21.6				
Q51	(0) Missing	8	17.8	--	37	2.811	1.175	4
	(1) Not Effective	8	17.8	21.6				
	(2) Somewhat Effective	5	11.1	13.5				
	(3) Moderately Effective	10	22.2	27.0				
	(4) Very Effective	14	31.1	37.8				
Q52	(0) Missing	8	17.8	--	37	2.730	1.122	3
	(1) Not Effective	8	17.8	21.6				

TABLE VIII (Continued)

FREQUENCY DISTRIBUTION AND DESCRIPTIVE STATISTICS FOR HORIZONTAL ALLOCATION QUESTIONS

Variable*	Response Group	Absolute Frequency	Relative Frequency (Percent)	Adjusted Frequency (Percent)	Valid Cases	Mean	Standard Deviation	Mode
	(2) Somewhat Effective	5	11.1	13.5				
	(3) Moderately Effective	13	28.9	35.1				
	(4) Very Effective	11	24.4	29.7				
Q53	(0) Missing	7	15.6	--	38	2.211	1.189	1
	(1) Not Effective	15	33.3	39.5				
	(2) Somewhat Effective	8	17.8	21.1				
	(3) Moderately Effective	7	15.6	18.4				
	(4) Very Effective	8	17.8	21.1				
Q54	(0) Missing	7	15.6	--	38	2.684	1.042	2
	(1) Not Effective	5	11.1	13.2				
	(2) Somewhat Effective	13	28.9	34.2				
	(3) Moderately Effective	9	20.0	23.7				
	(4) Very Effective	11	24.4	28.9				
Q55	(0) Missing	5	11.1	--	40	1.400	0.496	1
	(1) Yes	16	35.5	40.0				
	(2) No	24	53.3	60.0				

TABLE VIII (Continued)

FREQUENCY DISTRIBUTION AND DESCRIPTIVE STATISTICS FOR HORIZONTAL ALLOCATION QUESTIONS

Variable*	Response Group	Absolute Frequency	Relative Frequency (Percent)	Adjusted Frequency (Percent)	Valid Cases	Mean	Standard Deviation	Mode
Q56	(0) Missing	8	17.6	--	37	1.162	0.374	1
	(1) Yes	31	68.9	83.8				
	(2) No	6	13.3	16.2				
Q57	(0) Missing	8	17.8	--	37	1.757	0.435	2
	(1) Yes	28	62.3	75.7				
	(2) No	9	20.0	24.3				

* Variable names are described in Table IV, Appendix F.

APPENDIX J

Tables of Bivariate Correlations

These tables present the significant linear correlations between each pair of variables defined in Appendix F. The sign indicates either a negative or positive relationship between the variables as coded in Appendices G, H, and I.

TABLE IX

BIVARIATE CORRELATIONS AMONG DEMOGRAPHICS

Demographic Variables[a]	Demographic Variables[a]				
	Q5	Q10	Q11	Q12	Q14
Q5	--	--	-.3295*	--	--
Q6	.4071**	--	--	.3162*	--
Q7	--	--	--	--	--
Q8	.3612*	--	--	--	--
Q9	--	.8872**	--	--	--
Q10	--	--	--	--	--
Q12	--	--	--	--	--
Q13	--	--	--	--	--
Q15	--	--	--	.3586*	--

TABLE IX (Continued)

BIVARIATE CORRELATIONS AMONG DEMOGRAPHICS

Demographic Variables[a]	Demographic Variables[a]				
	Q16	Q17	Q19	Q20	Q21
Q5	--	--	--	--	--
Q6	--	--	--	--	--
Q7	--	--	--	--	--
Q8	--	--	--	--	--
Q9	--	-.3111*	--	--	--
Q10	-.3219*	.3159*	--	--	--
Q12	.3182*	--	--	--	--
Q13	--	--	--	--	.3832**
Q15	--	--	--	--	.3337*

TABLE IX (Continued)

BIVARIATE CORRELATIONS AMONG DEMOGRAPHICS

Demographic Variables[a]	Demographic Variables[a]				
	Q22	Q23	Q24	Q25	Q27
Q5	--	--	--	--	--
Q6	--	--	--	--	--
Q7	--	--	--	--	--
Q8	--	--	--	--	--
Q9	--	--	--	--	.6440***
Q10	--	--	--	--	.5464***
Q12	--	.3546*	--	.2984*	-.4147**
Q13	--	--	--	--	--
Q15	--	--	--	--	--

TABLE IX (Continued)

BIVARIATE CORRELATIONS AMONG DEMOGRAPHICS

Demographic Variables[a]	Demographic Variables[a]				
	Q5	Q10	Q11	Q12	Q14
Q16	--	-.3219*	--	.3182*	--
Q17	--	.3159*	--	--	--
Q18	--	--	--	--	-.3153*
Q22	--	--	--	--	--
Q23	--	--	--	.3546*	--
Q24	--	--	--	--	--
Q25	--	--	--	.2984*	--
Q26	--	--	--	--	--
Q27	--	.5464***	--	-.4147**	--

TABLE IX (Continued)

BIVARIATE CORRELATIONS AMONG DEMOGRAPHICS

Demographic Variables[a]	Demographic Variables[a]				
	Q16	Q17	Q19	Q20	Q21
Q16	--	.5082***	--	--	--
Q17	.5082***	--	--	--	--
Q18	--	--	--	--	--
Q22	.3120*	--	--	.5171***	.3530*
Q23	--	.3153*	--	.3592*	.5036***
Q24	--	--	.3970**	--	.5151***
Q25	--	--	--	--	.3050*
Q26	--	--	--	-.5483***	--
Q27	--	--	--	--	--

TABLE IX (Continued)

BIVARIATE CORRELATIONS AMONG DEMOGRAPHICS

Demographic Variables[a]	Demographic Variables[a]				
	Q22	Q23	Q24	Q25	Q27
Q16	.3120*	--	--	--	--
Q17	--	.3153*	--	--	--
Q18	--	--	--	--	--
Q22	--	.3801**	--	--	--
Q23	.3801**	--	.5046***	.3974**	--
Q24	--	.5036***	--	.4929***	--
Q25	--	.3974**	.4929***	--	-.3325*
Q26	--	--	--	--	--
Q27	--	--	--	-.3325*	--

[a] Variable names are defined in Table IV, Appendix F.

*p≤.05
**p≤.01
***p≤.001

TABLE X

BIVARIATE CORRELATIONS AMONG PROGRAM CHARACTERISTICS

Program Characteristic Variables[a]	Program Characteristic Variables[a]			
	Q29	Q30	Q31	Q33
Q28	.4902**	.3541*	.4428*	.6157***
Q29	--	--	.7500***	--
Q30	--	--	.6213***	.4281*
Q31	--	--	--	--
Q32	--	--	--	--
Q33	--	--	--	--
Q34	--	--	--	--
Q36	--	--	--	--
Q40	--	.3722*	.4260*	--

TABLE X (Continued)

BIVARIATE CORRELATIONS AMONG PROGRAM CHARACTERISTICS

Program Characteristic Variables[a]	Program Characteristic Variables[a]		
	Q34	Q38	Q39
Q28	--	--	--
Q29	--	--	--
Q30	--	--	--
Q31	--	--	-.5492**
Q32	.5271*	--	--
Q33	--	--	--
Q34	--	--	--
Q36	-.4677*	--	--
Q40	.4752**	-.5739**	-.5164***

[a] Variable names are defined in Table IV, Appendix F.

*p≤.05
**p≤.01
***p≤.001

TABLE XI

BIVARIATE CORRELATIONS BETWEEN DEMOGRAPHICS AND PROGRAM CHARACTERISTICS

Demographic Variables[a]	Program Characteristics Variables[a]						
	Q28	Q29	Q30	Q31	Q32	Q33	Q34
Q5	--	--	-.3700*	--	--	-.6855***	--
Q7	--	--	-.3938*	-.4475*	--	--	--
Q8	--	--	--	--	-.5490*	--	--
Q16	--	--	-.3564*	--	--	--	--
Q20	--	--	-.3406*	--	--	-.4338*	--
Q22	-.3589*	--	--	--	--	-.4404*	--
Q23	--	-.4000*	--	--	--	--	--
Q26	--	--	--	--	-.2404***	--	-.3488*

TABLE XI (Continued)

BIVARIATE CORRELATIONS BETWEEN DEMOGRAPHICS AND PROGRAM CHARACTERISTICS

Demographic Variables[a]	Program Characteristics Variables[a]		
	Q38	Q39	Q40
Q10	--	-.3321*	--
Q11	--	.3830*	--
Q12	--	.3846*	--
Q14	.4572**	--	--
Q15	--	.3315*	-.4264**

[a] Variable names are defined in Table IV, Appendix F.

*$p \leq .05$
**$p \leq .01$
***$p \leq .001$

TABLE XII

BIVARIATE CORRELATION[a] AMONG PERCEIVED EFFECTIVENESS VARIABLES

Perceived Effectiveness Variables[b]	Perceived Effectiveness Variables[b]					
	Q43	Q44	Q45	Q46	Q47	Q48
Q43	--	.7325	.3569**	.3755*	.4060*	.4127**
Q44	--	--	--	--	.3251*	--
Q45	--	--	--	.5513	.7340	.8237
Q46	--	--	--	--	.6730	.4552**
Q47	--	--	--	--	--	.5960
Q48	--	--	--	--	--	--

TABLE XII (Continued)

BIVARIATE CORRELATION[a] AMONG PERCEIVED EFFECTIVENESS VARIABLES

Perceived Effectiveness Variables[b]	Perceived Effectiveness Variables[b]					
	Q49	Q50	Q51	Q52	Q53	Q54
Q43	.6379	.5415	--	--	--	--
Q44	.3756*	.3691*	.4968**	.3699*	.6160	.3453*
Q45	.6782	.6760	.4515**	.3318*	.4955**	--
Q46	.6562	.6644	.3807*	.5144	.3392*	.4097*
Q47	.7571	.6017	.4389**	.3704*	--	.3994*
Q48	.6444	.6147	.4032*	--	.3838*	--

TABLE XII (Continued)

BIVARIATE CORRELATION[a] AMONG PERCEIVED EFFECTIVENESS VARIABLES

Perceived Effectiveness Variables[b]	Perceived Effectiveness Variables[b]					
	Q49	Q50	Q51	Q52	Q53	Q54
Q49	--	.7697	.4295**	--	.6167	.4316**
Q50	.7697	--	.4632**	.4067*	.5221	.3807**
Q51	.4295**	--	--	.4450**	--	--
Q52	--	--	--	--	--	.4670**
Q53	.6167	--	--	--	--	.3821*
Q54	.4316**	--	--	--	--	--

[a] All correlations are at the p≤.001 significance level unless noted as follows:
*p≤.05
**p≤.01

[b] Variable names are defined in Table IV, Appendix F.

TABLE XIII

BIVARIATE CORRELATIONS BETWEEN DEMOGRAPHICS AND THE PERCEIVED EFFECTIVENESS

Demographic Variables[a]	Perceived Effectiveness Variables[a]				
	Q43	Q44	Q45	Q48	Q49
Q7	--	--	--	--	--
Q11	.4076*	--	--	--	--
Q12	--	--	--	--	--
Q14	--	--	-.3211*	-.3868*	--
Q15	--	--	--	--	--
Q17	--	--	--	--	--
Q19	--	.3513*	--	--	--
Q20	--	--	-.3855	--	--
Q26	.3731*	--	.5210***	.5777***	--
Q27	--	--	--	--	-.5293

TABLE XIII (Continued)

BIVARIATE CORRELATIONS BETWEEN DEMOGRAPHICS AND THE PERCEIVED EFFECTIVENESS

Demographic Variables[a]	Perceived Effectiveness Variables[a]			
	Q50	Q53	Q54	Q55
Q7	--	--	-.4820**	--
Q11	--	.3471*	--	--
Q12	--	--	-.4117**	--
Q14	--	--	--	--
Q15	--	-.3748*	--	--
Q17	--	-.3197*	--	--
Q19	--	--	--	--
Q20	--	--	--	-.4155**
Q26	.3631*	--	--	.3487*
Q27	--	--	--	--

[a] Variable names are defined in Table IV, Appendix F.

*$p \leq .05$

**$p \leq .01$

***$p \leq .001$

TABLE XIV

BIVARIATE CORRELATIONS BETWEEN PROGRAM CHARACTERISTICS AND PERCEIVED EFFECTIVENESS

Program Characteristic Variables[a]	Perceived Effectiveness Variables[a]					
	Q45	Q46	Q47	Q48	Q49	Q50
Q29	--	--	--	--	--	--
Q30	--	--	.4223*	--	.5891***	.5368**
Q31	--	--	--	--	--	--
Q32	-.5062*	--	-.6861**	--	--	--
Q36	--	.4247*	--	--	--	--
Q38	--	--	--	--	--	--
Q39	--	--	--	--	--	--
Q40	--	--	--	--	--	--

TABLE XIV (Continued)

BIVARIATE CORRELATIONS BETWEEN PROGRAM CHARACTERISTICS AND PERCEIVED EFFECTIVENESS

Program Characteristic Variables[a]	Perceived Effectiveness Variables[a]				
	Q51	Q52	Q53	Q54	Q56
Q29	--	--	--	--	-.3934*
Q30	--	--	.4607**	.5940***	--
Q31	--	--	--	.5461**	--
Q32	--	--	--	--	--
Q36	--	--	--	--	--
Q38	--	--	--	-.4236*	--
Q39	-.3989*	--	--	--	--
Q40	--	.3917*	--	.4662**	--

[a] Variable names are defined in Table IV, Appendix F.

*$p \leq .05$

**$p \leq .01$

***$p \leq .001$

TABLE XV

BIVARIATE CORRELATIONS BETWEEN
DEMOGRAPHICS/PROGRAM CHARACTERISTICS
AND THE UTILITY OF HORIZONTAL ALLOCATION

Program Characteristics Variables[a]	Utility Variables[a]	
	Q42	Q57
Q26	-.6172***	-.5239***
Q30	.4884**	.4804**
Q31	.4159*	.4608**
Q32	-.5291*	--
Q39	-.3480*	--

[a] Variable names are described in Table IV, Appendix F.
*p≤.05
**p≤.01
***p≤.001

TABLE XVI

BIVARIATE CORRELATIONS
BETWEEN THE PERCEIVED EFFECTIVENESS AND
UTILITY OF HORIZONTAL ALLOCATION

Perceived Effectiveness Variables[a]	Utility Variables[a]	
	Q42	Q57
Q43	.4838**	.6914***
Q44	.4352**	.5845***
Q45	.4228**	.4549**
Q46	.5384***	.4337**
Q47	.4352**	.3518*
Q48	.4349**	.4792**
Q49	.5381***	.5974***

TABLE XVI (Continued)

BIVARIATE CORRELATIONS
BETWEEN THE PERCEIVED EFFECTIVENESS AND
UTILITY OF HORIZONTAL ALLOCATION

Perceived Effectiveness Variables[a]	Utility Variables[a]	
	Q42	Q57
Q50	.4704**	.5790***
Q51	--	.4090*
Q52	.3809*	.4783**
Q53	.4324**	.4773**
Q54	.3821*	.3698*
Q55	--	.3745*

[a] Variable names are described in Table IV, Appendix F.

*$p \leq .05$

**$p \leq .01$

***$p \leq .001$

TABLE XVII

INTERCORRELATIONS OF NOMINAL SCALE VARIABLES AND HORIZONTAL ALLOCATION UTILITY VARIABLES

Nominal Scale Variables[a]	Horizontal Allocation Utility Variables[a]	
	Q42	Q57
Q1	.2366	.5567
Q2	.5683	.1394
Q3	.1703	.8217
Q4	.3591	.5258
Q29	.1530	.3231
Q31	.1037	.0880
Q34	.7245	.9373
Q36	.9007	.7750
Q38	.8232	.7343

[a] Variable names are defined in Table IV, Appendix F.

APPENDIX K

System Development/Acquisition Programs Represented in the Interviews

TABLE XVIII

SYSTEM DEVELOPMENT/ACQUISITION PROGRAMS REPRESENTED IN THE INTERVIEWS

Program ID	Type of System
AEGIS	Fleet Air Defense System
AFEES	Air Force Entrance Examination System
ALQ 131	ECM Pod For Aeronautical Systems
ATEC	Ground-based Communications Monitoring System
AWACS	Airborne Warning and Control System
B-52 OAS	Offensive Avionics System
BETA	Tactical Intelligence Data System
---	Ballistic Missile Defense System Technology Program
BMEWS TOR	Tactical Operations Room Upgrade
CCPDS	Command and Control System
CONUS OTH	Over the Horizon Radar System
DMSP	Defense Meteorological Satellite Program
DSP	Command and Control System
F-15 AISF	Avionics Integrated Support Facility
GEADGE	412L System Replacement
GEODSS	Deep Space Tracking System
GLCM	Ground Launched Cruise Missile
ICAM	Industry Manufacturing Productivity Program
JSS	Command and Control System
JTIDS	Command, Control and Communications System
---	NASA Viking Program
OASIS	Intelligence Processing System
PAVE PAWS	Command and Control System
Project Impact	Office Automation Program
SACDIN	Communications System
SAGE	Air Defense System
---	SAMSO Satellite Control Facility System

TABLE XVIII (Continued)

SYSTEM DEVELOPMENT/ACQUISITION PROGRAMS
REPRESENTED IN THE INTERVIEWS

Program ID	Type of System
SEEK IGLOO	Ground-based Radar System
SPADOC	Space Defense Operations Center System
SURTASS	Sonar Surveillance System
STEM (407L)	System Training and Exercise Module
TAC FIRE	Field Artillery Fire Detection Center System
TAC STADS	Joint Tactical Air Defense System
TCCF (478T)	Tactical Communication Control Facility
TIPI	Intelligence Display Control Storage and Retrieval System
---	TOS Follow-on System
329A	A-10 Aircraft System
MACIMS (416L)	Passenger Service and Cargo Automation Program
427M	Command and Control Upgrade Program
436M	WWMCCS Interface Program

APPENDIX L

Results of Significance Tests

TABLE XIX

RESULTS OF SIGNIFICANCE TESTS*

Variable	n	P	t	t_α	Result
Q39	36	.722	2.664	1.645	Reject
Q42	39	.821	4.010	1.645	Reject
Q43	38	.711	2.601	1.645	Reject
Q44	38	.710	2.589	1.645	Reject
Q45	38	.815	3.884	1.645	Reject
Q46	38	.789	3.563	1.645	Reject
Q47	38	.816	3.896	1.645	Reject
Q48	38	.842	4.216	1.645	Reject
Q49	37	.702	2.457	1.645	Reject
Q50	37	.675	2.129	1.645	Reject
Q51	37	.784	3.455	1.645	Reject
Q52	37	.784	3.455	1.645	Reject
Q53	38	.605	1.295	1.645	Fail to Reject
Q54	38	.868	4.537	1.645	Reject
Q55	40	.400	-0.791	1.645	Fail to Reject
Q57	37	.757	3.126	1.645	Reject

* The students' one-tailed t test was used to test the null (H_o) and alternate (H_a) hypotheses described below:

H_o: $P_o=0.50$

H_a: $P_o>0.50$

Test Statistic: $$t = \frac{P-P_o}{\frac{P_o(1-P_o)}{n}}$$

Significance Level: $\alpha=.05$

Degrees of Freedom(d.f.): n-1

Rejection Region: $t>t_{\alpha,d.f.}$

where

a. P_o is equal to the percentage of respondents who answered a question positively (i.e., yes for a dichotomous question, and at least somewhat effective for the perceived effectiveness questions).

b. n is the number of respondents answering that question.

c. Fail to Reject means that there is sufficient evidence to support the null hypothesis that P is equal to 0.50.

d. Reject means that the evidence supports the alternate hypothesis that P is greater than 0.50.

Vita

Virgil L. Cooper was born on 8 December 1946 in Jamestown, Tennessee. He graduated from the Alvin C. York Agricultural Institute in 1964, and enlisted in the U.S. Air Force in April 1966. After completing the Airborne Radio Repairman Course at the Keesler Technical Training Center, Keesler AFB, Mississippi, he was assigned to the Air Force Eastern Test Range at Patrick AFB, Florida in January 1967. During the next 5½ years, he served as a High Frequency Operator/Technician on the Apollo Range Instrumentation Aircraft (EC-135N).

In August 1972, Staff Sergeant Cooper was selected for the Airman Education and Commissioning Program, and transferred to Oklahoma State University where he received a Bachelor of Science degree in Mathematics in May 1974. After completing Officers Training School in August 1974, he was commissioned a Second Lieutenant and assigned to the Electronic Systems Division(ESD) of the Air Force Systems Command at Hanscom AFB, Massachusetts as a Computer Systems Analyst. After 9 months, he was transferred to the ESD Field Office in Colorado Springs, Colorado. During the next 3 years, he participated in all phases of the system acquisition cycle as a Computer Systems Analyst, and System Integration and Test Monitor on a major C^3 acquisition program.

In May 1978, he entered the Air Force Institute of Technology as a graduate student in Systems Management.

Permanent Address: Route 3, Box 361
Jamestown, Tennessee 38556